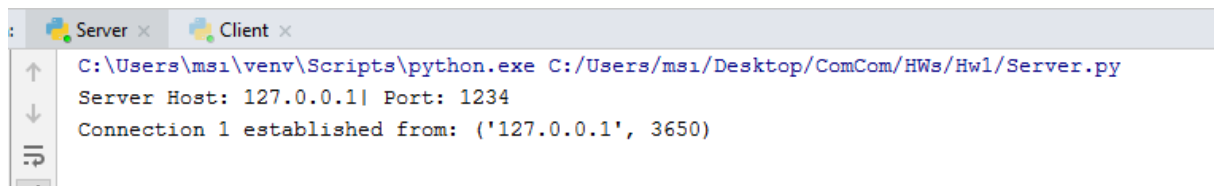


BLG 433E
COMPUTER COMMUNICATIONS
2019-2020 FALL / PROJECT 1
SOCKET PROGRAMMING

REPORT

BEHLÜL GÜLMEZ
150140113

Server.py shows connections

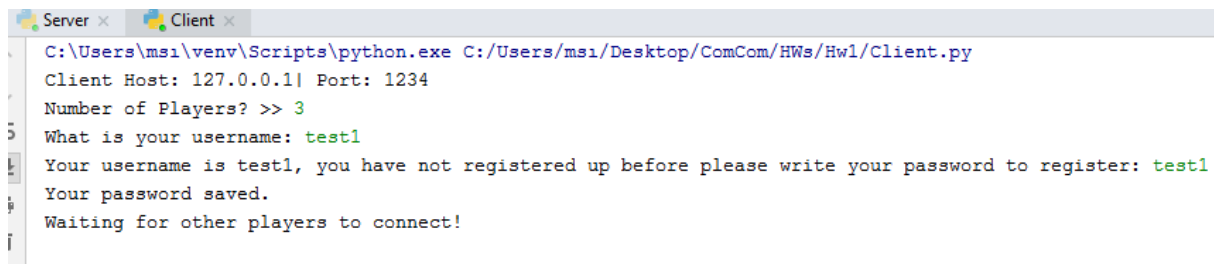


```
C:\Users\msi\venv\Scripts\python.exe C:/Users/msi/Desktop/ComCom/HWs/Hw1/Server.py
Server Host: 127.0.0.1| Port: 1234
Connection 1 established from: ('127.0.0.1', 3650)
```

Client.py takes user's info to register or login. Also takes number for how many players will play.

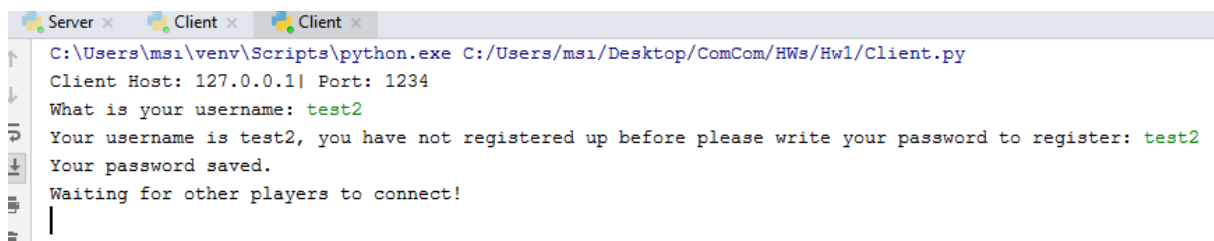
I am testing with 3.

This is client 1.



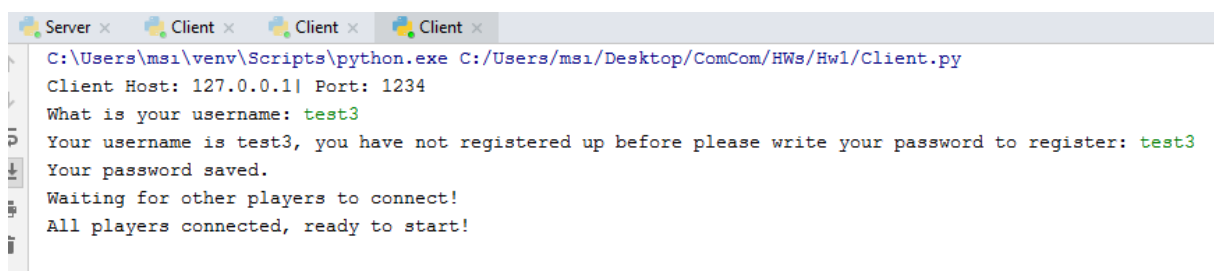
```
C:\Users\msi\venv\Scripts\python.exe C:/Users/msi/Desktop/ComCom/HWs/Hw1/Client.py
Client Host: 127.0.0.1| Port: 1234
Number of Players? >> 3
What is your username: test1
Your username is test1, you have not registered up before please write your password to register: test1
Your password saved.
Waiting for other players to connect!
```

This is client 2.



```
C:\Users\msi\venv\Scripts\python.exe C:/Users/msi/Desktop/ComCom/HWs/Hw1/Client.py
Client Host: 127.0.0.1| Port: 1234
What is your username: test2
Your username is test2, you have not registered up before please write your password to register: test2
Your password saved.
Waiting for other players to connect!
|
```

This is client 3.



```
C:\Users\msi\venv\Scripts\python.exe C:/Users/msi/Desktop/ComCom/HWs/Hw1/Client.py
Client Host: 127.0.0.1| Port: 1234
What is your username: test3
Your username is test3, you have not registered up before please write your password to register: test3
Your password saved.
Waiting for other players to connect!
All players connected, ready to start!
```

After everyone show up game start. Server asks clients for letter in order. Firstly client1.

```
Server x Client x Client x Client x
Client Host: 127.0.0.1 Port: 1234
Number of Players? >> 3
What is your username: test1
Your username is test1, you have not registered up before please write your password to register: test1
Your password saved.
Waiting for other players to connect!
All players connected, ready to start!
Your Turn!
You have 7 attempts_left

-----
Incorrect Guesses List:

Guess a Letter:
```

After make a guess next player will play.

```
Server x Client x Client x Client x

-----
Incorrect Guesses List:

Guess a Letter: a
Correct Letter!
_ a _ _ _ _
Incorrect Guesses List:

Waiting for other players to play...
You have 7 attempts_left

Player 2s turn...
```

If guess wrong, attempts_left -= 1

```
Server x Client x Client x Client x
_ a _ _ _ _
Incorrect Guesses List:

Guess a Letter: e
Incorrect Letter!
_ a _ _ _ _
Incorrect Guesses List: e

Waiting for other players to play...
You have 6 attempts_left

Player 3s turn...
```

Player can guess a word too.

```
Server x Client x Client x Client x
_ a _ _ _ _
Incorrect Guesses List: e

Guess a Letter: test
Incorrect Answer!
_ a _ _ _ _
Incorrect Guesses List: e test

Waiting for other players to play...
You have 5 attempts_left

Player 1s turn...
```

If one player guess correctly clients close.

```
Server x Client x Client x Client x

Guess a Letter: famous
Correct Answer!
f a m o u s
Incorrect Guesses List: e test

f a m o u s
Incorrect Guesses List: e test

You Win! Game Over!
Game Over!

Process finished with exit code 0
|
```

If attempts_left over clients close too.

```
Server x Client x Client x Client x

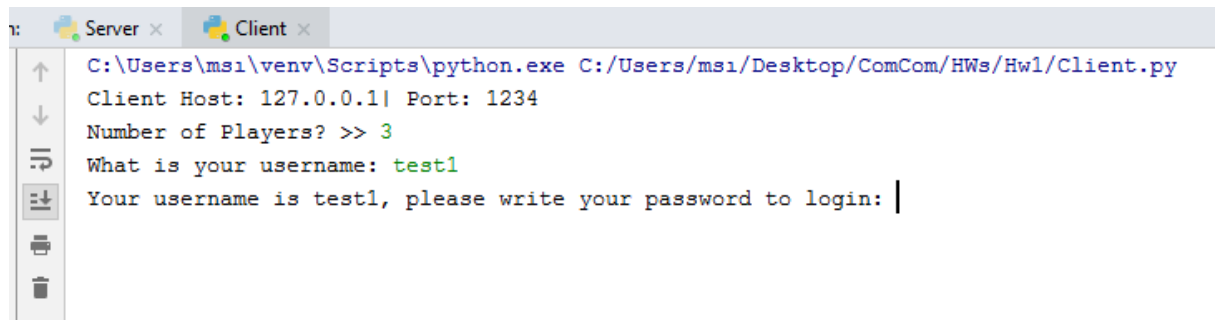
Guess a Letter: o
Incorrect Letter!
_ _ e e _ e
Incorrect Guesses List: q w r t y u o

_ _ e e _ e
Incorrect Guesses List: q w r t y u o

You Lose! Game Over!
Game Over!

Process finished with exit code 0
|
```

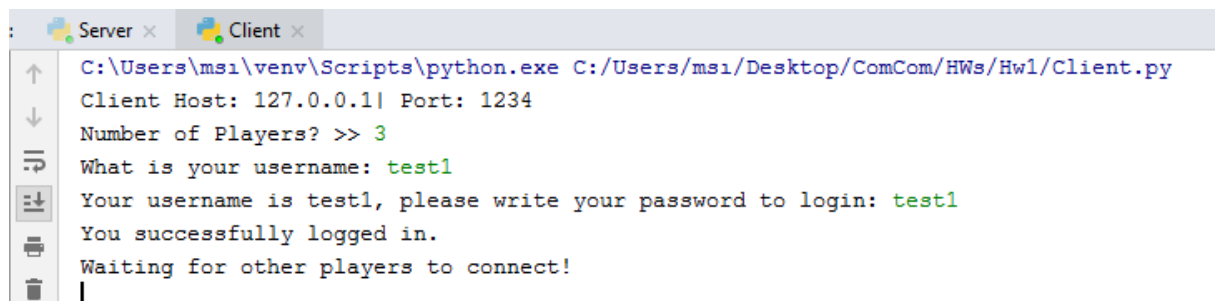
If same username tries to enter again. server asks for password to login.



A screenshot of a terminal window with two tabs: 'Server' and 'Client'. The 'Client' tab is active, showing the execution of a Python script. The output text is as follows:

```
C:\Users\msi\venv\Scripts\python.exe C:/Users/msi/Desktop/ComCom/HWs/Hw1/Client.py
Client Host: 127.0.0.1| Port: 1234
Number of Players? >> 3
What is your username: test1
Your username is test1, please write your password to login: |
```

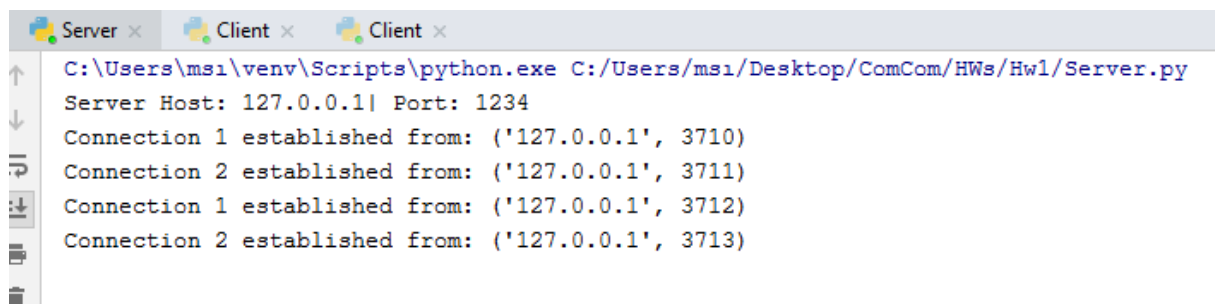
If player writes true password, server accepts.



A screenshot of a terminal window with two tabs: 'Server' and 'Client'. The 'Client' tab is active, showing the execution of a Python script. The output text is as follows:

```
C:\Users\msi\venv\Scripts\python.exe C:/Users/msi/Desktop/ComCom/HWs/Hw1/Client.py
Client Host: 127.0.0.1| Port: 1234
Number of Players? >> 3
What is your username: test1
Your username is test1, please write your password to login: test1
You successfully logged in.
Waiting for other players to connect!
|
```

After clients close and new clients join to server, connection numbers starts form 1 again.



A screenshot of a terminal window with three tabs: 'Server', 'Client', and 'Client'. The 'Server' tab is active, showing the execution of a Python script. The output text is as follows:

```
C:\Users\msi\venv\Scripts\python.exe C:/Users/msi/Desktop/ComCom/HWs/Hw1/Server.py
Server Host: 127.0.0.1| Port: 1234
Connection 1 established from: ('127.0.0.1', 3710)
Connection 2 established from: ('127.0.0.1', 3711)
Connection 1 established from: ('127.0.0.1', 3712)
Connection 2 established from: ('127.0.0.1', 3713)
```

This class for game info and datas. Holds incorrect letters ect.

```
Server.py x Client.py x
19
20 class GameClass:
21     word = ""
22     gameString = ""
23     numberOfIncorrectGuesses = 0
24     listOfIncorrectLetters = []
25     turn = 1
26     attempts_left = 7
27
28     def __init__(self, word):
29         self.listOfIncorrectLetters = []
30         self.word = word
31         for i in range(len(word)):
32             self.gameString += "_"
33
34     def guessLetter(self, letterGuessed):
35         if len(letterGuessed) > 1:
36             if letterGuessed == self.word:
37                 self.gameString = self.word
38                 return 'Correct Answer!'
39             else:
40                 self.numberOfIncorrectGuesses += 1
41                 self.attempts_left -= 1
42                 self.listOfIncorrectLetters.append(letterGuessed)
43                 return 'Incorrect Answer!'
44         else:
45             if letterGuessed in self.listOfIncorrectLetters or letterGuessed in self.gameString:
46                 self.numberOfIncorrectGuesses += 1
47                 self.attempts_left -= 1
48                 return 'Incorrect Letter!'
49             elif letterGuessed not in self.word:
50                 self.numberOfIncorrectGuesses += 1
51                 self.attempts_left -= 1
52                 self.listOfIncorrectLetters.append(letterGuessed)
53                 return 'Incorrect Letter!'
54             else:
55                 gameString = list(self.gameString)
56                 for i in range(len(self.word)):
57                     if self.word[i] == letterGuessed:
58                         gameString[i] = letterGuessed
59                 self.gameString = ''.join(gameString)
60                 return 'Correct Letter!'
61
62     def isOver(self):
```

This functions helps for send data to client.

```
def sendMessage(c, message):
    data = bytes([len(message)]) + bytes(message, 'utf8')
    c.send(data)

def sendDataForGuessLetter(c, game):
    Flag = 0
    data = bytes(game.gameString + " ".join(game.listOfIncorrectLetters), 'utf8')
    finalData = bytes([Flag]) + bytes([len(game.gameString)]) + bytes([len(" ".join(game.listOfIncorrectLetters))]) + data + bytes([game.numberOfIncorrectGuesses])
    c.send(finalData)
```

This is main game function. Until game is over, Works continuesly. Sends strings to clients and takes guesses from clients. Than compare them wirh actual word.

```
def multiplayerGame(c, player, game):
    global sign
    global full

    while True:
        currentTurn = game.turn
        while game.turn != player:
            if currentTurn != game.turn:
                sendMessage(c, " ".join(list(game.gameString)))
                sendMessage(c, "Incorrect Guesses List: " + " ".join(game.listOfIncorrectLetters) + "\n")
                sendMessage(c, "You have " + str(game.attempts_left) + " attempts_left" + "\n")
                sendMessage(c, 'Player ' + str(game.turn) + 's turn...')
                currentTurn = game.turn
            continue

        isOverCheck = game.isOver()
        if isOverCheck != '':
            sendDataForGuessLetter(c, game)
            sendMessage(c, isOverCheck)
            sendMessage(c, "Game Over!")
            game.changeTurn()
            break

        sendMessage(c, 'Your Turn!')
        sendMessage(c, "You have " + str(game.attempts_left) + " attempts_left" + "\n")

        sendDataForGuessLetter(c, game)

        receivedLetterLength = int(c.recv(1)[0])
        receivedLetter = c.recv(receivedLetterLength)
        letterGuessed = receivedLetter.decode('utf-8')

        sendMessage(c, game.guessLetter(letterGuessed))

        sendMessage(c, " ".join(list(game.gameString)))
        sendMessage(c, "Incorrect Guesses List: " + " ".join(game.listOfIncorrectLetters) + "\n")

        isOverCheck = game.isOver()
        if isOverCheck != '':
            sendDataForGuessLetter(c, game)
            sendMessage(c, isOverCheck)
            sendMessage(c, "Game Over!")
            game.changeTurn()
```


This function is where thread start and take username password from clients.

```
def Thread(c, sign):
    global totalNumberOfPlayers
    global full
    global game

    player = sign
    if player == 1:
        word = word_list[random.randint(0, 24)]
        game = GameClass(word)

    sendMessage(c, 'What is your username: ')
    lengthOfText = int(c.recv(1)[0])
    userName = c.recv(lengthOfText).decode('utf8')

    if userName not in user_password:
        registerChecker = 1
        c.send(bytes([registerChecker]))
        sendMessage(c, 'Your username is ' + userName + ', you have not registered up before please write your password to register: ')
        lengthOfText = int(c.recv(1)[0])
        text = c.recv(lengthOfText).decode('utf8')
        user_password[userName] = text
        sendMessage(c, 'Your password saved.')
    else:
        registerChecker = 0
        c.send(bytes([registerChecker]))
        sendMessage(c, 'Your username is ' + userName + ', please write your password to login: ')
        lengthOfText = int(c.recv(1)[0])
        text = c.recv(lengthOfText).decode('utf8')
        if user_password[userName] == text:
            passwordChecker = 1
            c.send(bytes([passwordChecker]))
        else:
            passwordChecker = 0
            c.send(bytes([passwordChecker]))
            sign -= 1
            if sign == 0:
                full = False
            c.close()

    sendMessage(c, 'Waiting for other players to connect!')
```

Main function of server.py.

```
if __name__ == '__main__':
    ip = "127.0.0.1"
    port = 1234

    serverSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    print('Server Host: ' + ip + ' | Port: ' + str(port))

    try:
        serverSocket.bind((ip, port))
    except socket.error as e:
        print(str(e))
    serverSocket.listen()

    c, address = serverSocket.accept()
    print("Connection " + str(sign + 1) + " established from: " + str(address))
    sign += 1

    c.send(bytes([sign]))
    totalNumberOfPlayers = int(c.recv(10).decode('utf-8'))

    start_new_thread(Thread, (c, sign))

while True:
    while sign < totalNumberOfPlayers:
        c, address = serverSocket.accept()
        print("Connection " + str(sign + 1) + " established from: " + str(address))
        sign += 1
        c.send(bytes([sign]))
        if sign == 1:
            totalNumberOfPlayers = int(c.recv(10).decode('utf-8'))
            start_new_thread(Thread, (c, sign))
    continue
```