

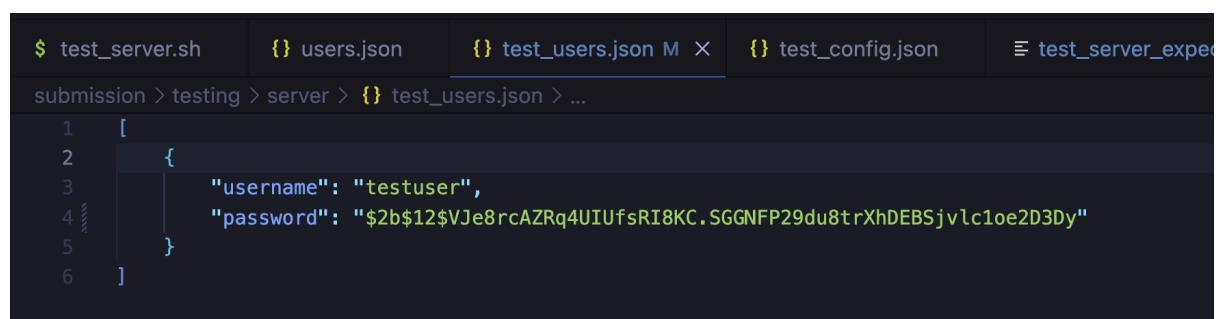
Bash Script

Non-Game Related Moves

To test the server, I created a Bash script to simulate client interactions via ncat while running server.py. The script reads commands from a .in file, sends them to the server through ncat, and compares the output with expected results, covering most scenarios except game play. It also uses a dedicated configuration file (test_config.json) and userdb (test_users.json).



```
server.sh  {} users.json  {} test_users.json M  {} test_config.json X  ⌂
submission > testing > server > {} test_config.json > ...
1  ↵ {
2    "port": 6507,
3    "userDatabase": "testing/server/test_users.json"
4 }
```

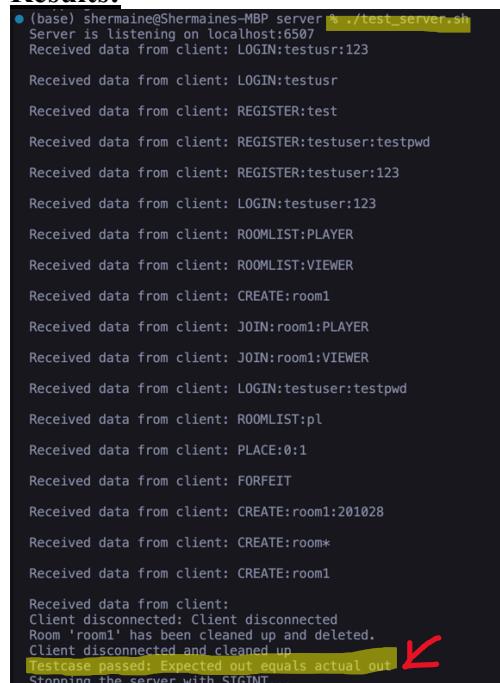


```
$ test_server.sh  {} users.json  {} test_users.json M ×  {} test_config.json  ⌂ test_server_expected
submission > testing > server > {} test_users.json > ...
1 [
2   {
3     "username": "testuser",
4     "password": "$2b$12$VJe8rcAZRq4UIUfsRI8KC.SGGNFP29du8trXhDEBSjvlc1oe2D3Dy"
5   }
6 ]
```

To execute the tests:

1. cd testing/server.
2. chmod +x test_server.sh.
3. Run ./test_server.sh.

Results:



```
(base) shermaine@Shermaine-MBP server % ./test_server.sh
Server is listening on localhost:6507
Received data from client: LOGIN:testusr:123
Received data from client: LOGIN:testusr
Received data from client: REGISTER:test
Received data from client: REGISTER:testuser:testpwd
Received data from client: REGISTER:testuser:123
Received data from client: LOGIN:testuser:123
Received data from client: ROOMLIST:PLAYER
Received data from client: ROOMLIST:VIEWER
Received data from client: CREATE:room1
Received data from client: JOIN:room1:PLAYER
Received data from client: JOIN:room1:VIEWER
Received data from client: LOGIN:testuser:testpwd
Received data from client: ROOMLIST:pl
Received data from client: PLACE:0:1
Received data from client: FORFEIT
Received data from client: CREATE:room1:201028
Received data from client: CREATE:room*
Received data from client: CREATE:room1
Received data from client:
Client disconnected: Client disconnected
Room 'room1' has been cleaned up and deleted.
Client disconnected and cleaned up
Testcase passed: Expected out equals actual out ↴
Stopping the server with SIGINT...
```

Tested the following in test script (test_server.in):

Login: non-existent user, invalid message format, wrong password, and successful login after registration.

Register: invalid registration format, successful registration, and attempting to register an existing user.

Authentication Handling: Verifying access to authenticated-only commands (ROOMLIST, CREATE, JOIN) results in a **BADAUTH** response if not authenticated.

Post Authentication

Roomlist: Testing invalid input modes and receiving appropriate error responses.

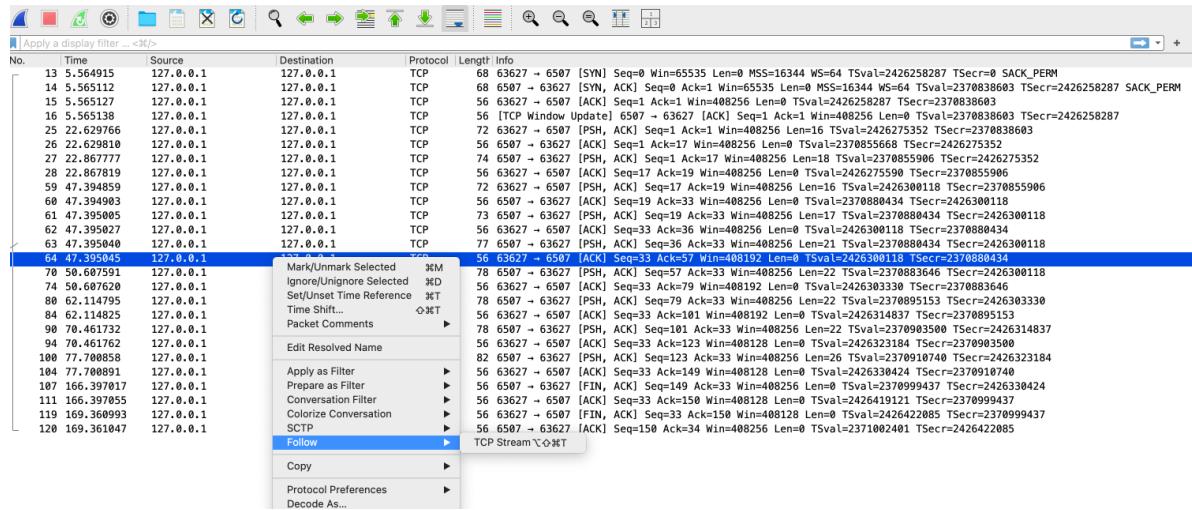
Game-Related Messages: Sending game commands (**PLACE**, **FORFEIT**) without joining a game returns a **NOROOM** response.

CREATE: Invalid format, non-acceptable characters('*'), and successful room creation.

See test_server.out for server responses to the commands sent from .in file

Wireshark testing

I used Wireshark to capture and analyse network traffic during some tests. The .pcapng file is included in the submission (testing/wireshark/info1112_assignment2_1.pcapng.gz), which can be opened using Wireshark to review and track the packet exchanges between the client and server. Right click on any of the entries and select follow > TCP stream to see packet exchange.



Terminal testing.

1. ran client.py, server.py concurrently
2. nc at as a server/client while running client.py/server.py

Note: When testing server.py and client.py concurrently, my server doesn't explicitly print the responses it is sending, it is implicit that they are transmitted in correct format, if not client would not be able to correctly handle and display those responses.

Scenario 1 – Client response Ackstatus received from server (Server-side Validation).

<pre>(base) shermaine@Shermaines-MBP submission % python3 client .py localhost 6507 LOGIN Enter Username: dummytest Enter password: 123 Error: User dummytest not found REGISTER Enter username: dummytest Enter password: 123 Successfully created user account dummytest REGISTER Enter username: dummytest Enter password: 1234 Error: User dummytest already exists ROOMLIST Do you want to have a room list as player or viewer? (player/viewer) PLAYER Error: You must be logged in to perform this action CREATE Enter room name you want to create: room1 Error: You must be logged in to perform this action JOIN Enter room name to join: room1 Join as (Player/Viewer): PLAYER Error: You must be logged in to perform this action JOIN Enter room name to join: room1 Join as (Player/Viewer): VIEWER Error: You must be logged in to perform this action LOGIN Enter Username: dummytest Enter password: 123 Welcome dummytest ROOMLIST Do you want to have a room list as player or viewer? (player/viewer) pl Error: Please input a valid mode. CREATE Enter room name you want to create: room*** Error: room*** is invalid CREATE Enter room name you want to create: dummyroom Successfully created room dummyroom Waiting for other player... </pre>	<pre>(base) shermaine@Shermaines-MBP submission % python3 server.py config.json Server is listening on localhost:6507 Received data from client: LOGIN:dummytest:123 Received data from client: REGISTER:dummytest:1234 Received data from client: ROOMLIST:PLAYER Received data from client: CREATE:room1 Received data from client: JOIN:room1:PLAYER Received data from client: JOIN:room1:VIEWER Received data from client: LOGIN:dummytest:123 Received data from client: ROOMLIST:pl Received data from client: CREATE:room*** Received data from client: CREATE:dummyroom</pre>
---	--

Scenario 2 - all clients are authenticated:

This scenario tests room management and player interactions: creating rooms, handling the max room limit(P3,SUP), listing available rooms based on player/viewer modes, joining rooms, and handling errors for player filled rooms or duplicate room creation. It also verifies that the room creator is prompted to make the first move and ensures proper error responses for unauthenticated actions.

Note: I temporarily set max_rooms to 2 (actual 256) for testing. With 2 rooms already created, attempts to create room3 failed, simulating the actual scenario. The client response to Ackstatus.CR_MAX_ROOM_CAP.value remained unchanged.

```
class Server:
    def __init__(self, config):
        self.config = config
        self.sel = selectors.DefaultSelector()
        self.port = config.port
        self.users = config.load_user_db()
        self.host = "localhost"
        self.server_socket = None
        self.authenticated_clients: Dict[socket.socket, Player] = {}
        self.rooms: Dict[str, Room] = {}
        self.max_rooms = 2 — actual 256
```

In-Game Related Scenarios (server and client test)

These scenarios demonstrate successful room deletion, player reset, and proper handling of GAMEEND messages after each game. Mid-game details are omitted for conciseness. Viewers leave only once the game ends and can re-join rooms as players or viewers. It highlights accurate turn switching, room management, role assignment ('X' and 'O'), and a viewer transitioning to a player in a new room (1.4). All actions occur **within the same client/server session**, showing the ability to re-join or create new rooms post-GAMEEND.

1.1 Won diagonally

```
from this
(base) shermanae@Shermanae-MBP:~/Desktop$ python3 client.py localhost 6507
LOGIN
Enter Username: SUP
Enter password: 123
Welcome hello
CREATE
Enter room name you want to create: rom
Successfully created room rom
Waiting for other player...
It's your turn. Please place marker
Type place to make a move, forfeit
to end game: PLACE
Col: 1
Row: 0
Validating move: row 0, col 1
Displaying board: 010000000
| | X | |
| | | |
| | | |
| | | |

It is the current player's turn
Type place to make a move, forfeit
to end game: PLACE
Col: 1
Row: 1
Validating move: row 1, col 1
Displaying board: 010020000
| | X | |
| | 0 | |
| | | |
| | | |

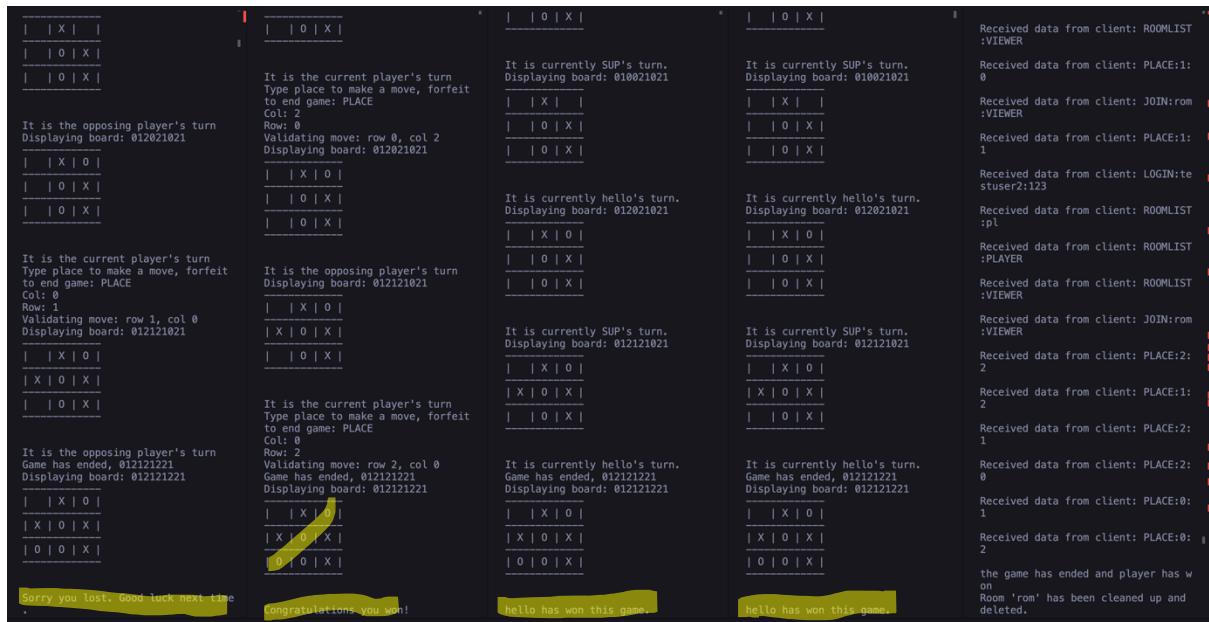
It is the opposing player's turn
Displaying board: 010020000
| | X | |
| | 0 | |
| | | |
| | | |

It is the current player's turn
Type place to make a move, forfeit
e: PLACE
Col: 1
Row: 1
Validating move: row 1, col 1
Invalid move, try again
Col: 1
Row: 2
Validating move: row 2, col 2
Displaying board: 010020001
| | X | |
| | 0 | |
| | | X |
| | | |

It is the opposing player's turn
Displaying board: 010020001
| | X | |
| | 0 | |
| | | X |
| | | |

It is currently SUP's turn.
Displaying board: 010020001
| | X | |
| | 0 | |
| | | X |
| | | |

It is currently hello's turn.
Displaying board: 010020021
| | X | |
| | 0 | |
| | | X |
| | | |
```



1.2 Won horizontally

```

CREATE
Enter room name you want to create: rom
Successfully created room rom
Waiting for other player...
It's your turn, Please place marker
Type place to make a move, forfeit
to end game: PLACE
Col: 0
Row: 1
Validating move: row 1, col 0
Displaying board: 000100000
| | |
| X | |
| | |
It is the opposing player's turn
Displaying board: 000100002
| | |
| X | |
| | 0 |

It is the current player's turn
Type place to make a move, forfeit
to end game: PLACE
Col: 1
Row: 1
Validating move: row 1, col 1
Displaying board: 000110002
| | |
| X | X |
| | 0 |

It is the opposing player's turn
Displaying board: 000110002
| | |
| X | X |
| | 0 |

It is the current player's turn
Type place to make a move, forfeit
to end game: PLACE
Col: 1
Row: 4
Validating move: row 4, col 1
Invalid move, try again
It is the opposing player's turn

JOIN
Enter room name to join: rom
Join as (Player/Viewer): PLAYER
Successfully joined room rom as a P
AYER
It is the opposing player's turn
Displaying board: 000100000
| | |
| X | |
| | |

It is the current player's turn
Type place to make a move, forfeit
to end game: PLACE
Col: 2
Row: 2
Validating move: row 2, col 2
Displaying board: 000100002
| | |
| X | |
| | 0 |

It is the opposing player's turn
Displaying board: 000110002
| | |
| X | X |
| | 0 |

It is the current player's turn
Type place to make a move, forfeit
to end game: PLACE
Col: 1
Row: 4
Validating move: row 4, col 1
Invalid move, try again
It is the opposing player's turn

JOIN
Enter room name to join: rom
Join as (Player/Viewer): VIEWER
Successfully joined room rom as a V
IEWER
Match between SUP and hello is curr
ently in progress, it is SUP's turn
Displaying board: 000100000
| | |
| X | |
| | |

It is currently hello's turn.
Displaying board: 000100002
| | |
| X | |
| | |

It is currently hello's turn.
Displaying board: 000100002
| | |
| X | |
| | |

It is currently SUP's turn.
Displaying board: 000110002
| | |
| X | X |
| | 0 |

It is currently hello's turn.
Displaying board: 000110002
| | |
| X | X |
| | 0 |

It is currently hello's turn.
Displaying board: 000110002
| | |
| X | X |
| | 0 |

```

The screenshot shows a terminal window with several text-based outputs representing a game session:

- Player 'hello' starts their turn.
- Player 'rom' makes a move at position (4,1).
- Player 'hello' makes a move at position (2,2).
- Player 'rom' makes a move at position (1,1).
- Player 'hello' makes a move at position (1,2).
- Player 'rom' makes a move at position (1,3).
- Player 'hello' makes a move at position (2,1).
- Player 'rom' makes a move at position (2,3).
- Player 'hello' makes a move at position (3,1).
- Player 'rom' makes a move at position (3,2).
- Player 'hello' makes a move at position (3,3).
- Player 'rom' makes a move at position (2,2).
- Player 'hello' makes a move at position (1,1).
- Player 'rom' makes a move at position (1,2).
- Player 'hello' makes a move at position (1,3).
- Player 'rom' makes a move at position (2,1).
- Player 'hello' makes a move at position (2,3).
- Player 'rom' makes a move at position (3,1).
- Player 'hello' makes a move at position (3,2).
- Player 'rom' makes a move at position (3,3).
- Player 'hello' wins the game.

1.3 Draw

```
Congratulations you won!
CREATE
Enter room name you want to create:
rom
Successfully created room rom
Waiting for other player...
It's your turn, Please place marker
Type place to make a move, forfeit
to end game: PLACE
Col: 0
Row: 0
Validating move: row 0, col 0
Displaying board: 100000000
| X |   |
|   |   |
|   |   |
|   |   |
|   |   |

It is the current player's turn
Type place to make a move, forfeit
to end game: PLACE
Col: 1
Row: 1
Validating move: row 1, col 1
Displaying board: 1000020000
| X |   |
|   |   |
|   |   |
|   |   |
|   |   |

It is the opposing player's turn
Displaying board: 100020000
| X |   |
|   |   |
|   |   |
|   |   |
|   |   |

It is the current player's turn
Type place to make a move, forfeit
to end game: PLACE
Col: 2
Row: 2
Validating move: row 2, col 2
Displaying board: 100020001
| X |   |
|   |   |
|   |   |
|   |   |
|   |   |

It is the current player's turn
Type place to make a move, forfeit
to end game: PLACE
Col: 1
Row: 2
Validating move: row 2, col 1
Displaying board: 100020021
| X |   |
|   |   |
|   |   |
|   |   |
|   |   |

It is the opposing player's turn
Displaying board: 100020001
JOIN
Enter room name to join: rom
Join as (Player/Viewer): PLAYER
Successfully joined room rom as a PAYER
It is the opposing player's turn
Displaying board: 100000000
| X |   |
|   |   |
|   |   |
|   |   |
|   |   |

SUP has won this game.
JOIN
Enter room name to join: rom
Join as (Player/Viewer): VIEWER
Successfully joined room rom as a VIEWER
Match between SUP and hello is curr
ently in progress, it is SUP's turn
Displaying board: 100000000
| X |   |
|   |   |
|   |   |
|   |   |
|   |   |

SUP has won this game.
JOIN
Enter room name to join: rom
Join as (Player/Viewer): VIEWER
Successfully joined room rom as a VIEWER
Match between SUP and hello is curr
ently in progress, it is SUP's turn
Displaying board: 100000000
| X |   |
|   |   |
|   |   |
|   |   |
|   |   |

Received data from client: JOIN:rom
:VIEWER
Received data from client: JOIN:rom
:VIEWER
Received data from client: PLACE:0
:0
Received data from client: PLACE:1
:1
Received data from client: PLACE:2
:2
Received data from client: PLACE:1
:2
Received data from client: PLACE:1
:0
Received data from client: PLACE:2
:0
Received data from client: PLACE:2
:1
Received data from client: PLACE:2
:2
Received data from client: PLACE:0
:1
Received data from client: PLACE:0
:0
Received data from client: PLACE:2
:0
Received data from client: PLACE:2
:1
Received data from client: PLACE:0
:1
Received data from client: PLACE:0
:2
Received data from client: CREATE:rom
:rom
Received data from client: JOIN:rom
:PLAYER
Received data from client: JOIN:rom
:VIEWER
Received data from client: PLACE:1
:0
Received data from client: PLACE:2
:1
```

It is the current player's turn Type place to make a move, forfeit to end game: PLACE Col: 2 Row: 2 Validating move: row 1, col 2 Displaying board: 112021021	X X 0 0 0 X 0 0 X	X X 0 0 0 X 0 0 X	It is currently hello's turn. Displaying board: 112020021
It is the opposing player's turn Displaying board: 112021021	X X 0 0 0 X 0 0 X	X X 0 0 0 X 0 0 X	It is currently hello's turn. Displaying board: 112020021
It is the current player's turn Displaying board: 112221021	X X 0 0 0 X 0 0 X	X X 0 0 0 X 0 0 X	It is currently SUP's turn. Displaying board: 112021021
It is the current player's turn Type place to make a move, forfeit to end game: PLACE Col: 0 Row: 1 Validating move: row 1, col 0 Displaying board: 112221021	X X 0 0 0 X 0 0 X	X X 0 0 0 X 0 0 X	It is currently SUP's turn. Displaying board: 112021021
It is the current player's turn Type place to make a move, forfeit to end game: PLACE Col: 0 Row: 2 Validating move: row 2, col 0 Game has ended: 112221121 Displaying board: 112221121	X X 0 0 0 X 0 0 X	X X 0 0 0 X 0 0 X	It is currently hello's turn. Displaying board: 112221021
Game ended in a draw.	X X 0 0 0 X 0 0 X	X X 0 0 0 X 0 0 X	It is currently hello's turn. Displaying board: 112221021
Game ended in a draw.	X X 0 0 0 X 0 0 X	X X 0 0 0 X 0 0 X	It is currently SUP's turn. Displaying board: 112221121
Game ended in a draw.	X X 0 0 0 X 0 0 X	X X 0 0 0 X 0 0 X	It is currently SUP's turn. Displaying board: 112221121
Game ended in a draw.	X X 0 0 0 X 0 0 X	X X 0 0 0 X 0 0 X	Room 'rom' has been cleaned up and deleted.

1.4 GAMEEND, viewer joins new room as a player

Scenario 2 – Out-of-turn moves

Out-of-turn moves are queued and executed when it becomes the player's turn, as seen with testuser's (P2) move at (2,2). An edge case occurs when dummytest (P1) tries to place at (1,1) out of turn, but the move is rejected because testuser (P2) had just made that move. Viewers (Clients 3 and 4) accurately reflect turn switching, with Client 3 joining before **BEGIN** and Client 4 joining **INPROGRESS**. All clients continue with other authenticated actions after the game ends.

Scenario 3 – Implicit forfeit

P2(testuser) forfeits via Ctrl+D(red arrow), all clients(viewers, players), are notified. dummytest wins by default.

```
(base) shermaine@Shermaines-MBP submission % python3 client.py localhost 6507
Enter Username: dummytest
Enter password: 123
Welcome dummytest
CREATE
Enter room name you want to create: room1
Successfully created room room1
Waiting for another player...
It's your turn, Please place marker
Type place to make a move, forfeit to end game
: PLACE
Col: 0
Row: 0
Validating move: row 0, col 0
Displaying board: 10000000
| X |   |
|   |   |
|   |   |
|   |   |

It is the opposing player's turn
Displaying board: 10000002
| X |   |
|   |   |
|   |   |
|   |   |

It is the current player's turn
Type place to make a move, forfeit to end game
: PLACE
Col: 2
Row: 2
Validating move: row 2, col 2
Displaying board: 10000002
| X |   |
|   |   |
|   |   |
|   |   |

It is the opposing player's turn
Displaying board: 10001002
| X |   |
|   |   |
|   |   |
|   |   |

It is the current player's turn
Type place to make a move, forfeit to end game
: PLACE
Col: 1
Row: 1
Validating move: row 1, col 1
Displaying board: 10001002
| X |   |
|   |   |
|   |   |
|   |   |

It is the opposing player's turn
Game has ended, 10001002
| X |   |
|   |   |
|   |   |
|   |   |

(base) shermaine@Shermaines-MBP sub
mission % python3 client.py localhost 6507
Enter Username: LOGIN
Enter Username: testuser
Enter password: 123
Welcome testuser
JOIN
Enter room name to join: room1
Join as (Player/Viewer): PLAYER
Successfully joined room room1 as a
PLAYER
It is the opposing player's turn
Displaying board: 10000000
| X |   |
|   |   |
|   |   |
|   |   |

It is the current player's turn
Type place to make a move, forfeit to end game
: PLACE
Col: 2
Row: 0
Validating move: row 0, col 0
Displaying board: 10000000
| X |   |
|   |   |
|   |   |
|   |   |

It is the opposing player's turn
Displaying board: 10000002
| X |   |
|   |   |
|   |   |
|   |   |

It is the current player's turn
Type place to make a move, forfeit to end game
: PLACE
Col: 2
Row: 1
Validating move: row 1, col 1
Displaying board: 10000002
| X |   |
|   |   |
|   |   |
|   |   |

It is the opposing player's turn
Displaying board: 10001002
| X |   |
|   |   |
|   |   |
|   |   |

It is the current player's turn
Type place to make a move, forfeit to end game
: PLACE
Col: 0
Row: 1
Validating move: row 1, col 0
Displaying board: 10001002
| X |   |
|   |   |
|   |   |
|   |   |

It is the opposing player's turn
Game has ended, 10001002
| X |   |
|   |   |
|   |   |
|   |   |

(base) shermaine@Shermaines-MBP su
mission % python3 client.py local
host 6507
LOGIN
Enter Username: hello
Enter password: 123
Welcome hello
JOIN
Enter room name to join: room2
Join as (Player/Viewer): VIEWER
Successfully joined room room1 as a
VIEWER
Match between dummytest and testus
er will commence, it is currently
dummytest's turn
Displaying board: 10000000
| X |   |
|   |   |
|   |   |
|   |   |

It is the current player's turn
Type place to make a move, forfeit to end game
: PLACE
Col: 2
Row: 2
Validating move: row 2, col 2
Displaying board: 10000002
| X |   |
|   |   |
|   |   |
|   |   |

It is the opposing player's turn
Displaying board: 10000002
| X |   |
|   |   |
|   |   |
|   |   |

It is the current player's turn
Type place to make a move, forfeit to end game
: PLACE
Col: 0
Row: 0
Validating move: row 0, col 0
Displaying board: 10000002
| X |   |
|   |   |
|   |   |
|   |   |

It is the opposing player's turn
Displaying board: 10001002
| X |   |
|   |   |
|   |   |
|   |   |

It is the current player's turn
Type place to make a move, forfeit to end game
: PLACE
Col: 2
Row: 1
Validating move: row 1, col 1
Displaying board: 10001002
| X |   |
|   |   |
|   |   |
|   |   |

It is the opposing player's turn
Displaying board: 10001002
| X |   |
|   |   |
|   |   |
|   |   |

It is the current player's turn
Type place to make a move, forfeit to end game
: PLACE
Col: 0
Row: 1
Validating move: row 1, col 0
Displaying board: 10001002
| X |   |
|   |   |
|   |   |
|   |   |

It is the opposing player's turn
Game has ended, 10001002
| X |   |
|   |   |
|   |   |
|   |   |

(base) shermaine@Shermaines-MBP su
mission % python3 client.py local
host 6507
LOGIN
Enter Username: SUP
Enter password: 123
Welcome SUP
JOIN
Enter room name to join: room2
Join as (Player/Viewer): VIEWER
Successfully joined room room1 as a
VIEWER
Match between testuser and dummyt
est is currently in progress, it is
testuser's turn
Displaying board: 10000002
| X |   |
|   |   |
|   |   |
|   |   |

It is the current player's turn
Type place to make a move, forfeit to end game
: PLACE
Col: 2
Row: 2
Validating move: row 2, col 2
Displaying board: 10000002
| X |   |
|   |   |
|   |   |
|   |   |

It is the opposing player's turn
Displaying board: 10000002
| X |   |
|   |   |
|   |   |
|   |   |

It is the current player's turn
Type place to make a move, forfeit to end game
: PLACE
Col: 0
Row: 0
Validating move: row 0, col 0
Displaying board: 10000002
| X |   |
|   |   |
|   |   |
|   |   |

It is the opposing player's turn
Displaying board: 10001002
| X |   |
|   |   |
|   |   |
|   |   |

It is the current player's turn
Type place to make a move, forfeit to end game
: PLACE
Col: 2
Row: 1
Validating move: row 1, col 1
Displaying board: 10001002
| X |   |
|   |   |
|   |   |
|   |   |

It is the opposing player's turn
Displaying board: 10001002
| X |   |
|   |   |
|   |   |
|   |   |

It is the current player's turn
Type place to make a move, forfeit to end game
: PLACE
Col: 0
Row: 1
Validating move: row 1, col 0
Displaying board: 10001002
| X |   |
|   |   |
|   |   |
|   |   |

It is the opposing player's turn
Game has ended, 10001002
| X |   |
|   |   |
|   |   |
|   |   |

(base) shermaine@Shermaines-MBP su
mission % python3 server.py config.js
Server is listening on localhost:6507
Received data from client: LOGIN:dum
mytest:123
Received data from client: LOGIN:t
estuser:123
Received data from client: LOGIN:hel
lo:123
Received data from client: LOGIN:SUP
:123
Received data from client: CREATE:ro
om1
Received data from client: JOIN:roo
m1:VIEWER
Received data from client: JOIN:roo
m1:PLAYER
Received data from client: PLACE:0:0
Received data from client: JOIN:roo
m1:PLAYER
Received data from client: PLACE:2:2
Received data from client: PLACE:1:1
Received data from client:
Client disconnected: Client disconne
cted
Room 'room1' has been cleaned up and
deleted.
Client disconnected and cleaned up
|
↑
server
still
running
```

```
It is the opposing player's turn
Game has ended, 100010002
Displaying board: 100010002

| X |   |
|   | X |
|   |   0 |

It is the opposing player's turn
Displaying board: 100010002

| X |   |
|   | X |
|   |   0 |

It is currently testuser's turn.
Game has ended, 100010002
Displaying board: 100010002

| X |   |
|   | X |
|   |   0 |

dummytest won due to the opposing
player forfeiting
□

dummytest won due to the opposing
player forfeiting
□

dummytest won due to the opposing
player forfeiting
□
```

Scenario 4 – Explicit forfeit

When dummytest (P1) sends FORFEIT message, testuser(P2) wins by default.

The image shows four terminal windows side-by-side, each displaying a different stage of a game session between two players, dummytest and testuser.

- Terminal 1:** Shows dummytest logging in and creating a room named 'rom'. It then sends a FORFEIT message, which is highlighted in yellow. The server responds with a message indicating testuser has won due to the opposing player's forfeiting.
- Terminal 2:** Shows testuser logging in and joining the room 'rom' as a viewer. The server responds with a message indicating dummytest has won due to the opposing player's forfeiting.
- Terminal 3:** Shows dummytest logging in and entering the room 'rom' as a viewer. The server responds with a message indicating testuser has won due to the opposing player's forfeiting.
- Terminal 4:** Shows dummytest logging in and sending a FORFEIT message. The server responds with a message indicating testuser has won due to the opposing player's forfeiting.

In both scenarios, the server handles the disconnection correctly, notifies all clients and viewers of the winner, deletes the room, keeps running and maintains all other client sessions active.

NC -l server to test client program

Test password/username length <= 20 (Client side validation)

The image shows two terminal windows. The left window shows a user attempting to register with a very long username and password, which results in an error message: "Username is too long, try again." The right window shows a user connecting to the server at port 1234 and successfully registering with the same long credentials.

A screenshot of a forum post with several comments. The post itself is a link to a GitHub issue. The first comment is from Chris Polak, who says: "Sorry no, you're correct, I think this was added later to the specs but no ACKSTATUS was created for it, so for this case, you can handle it client-side for REGISTER only :).". Below this, there are replies from Anonymous and Chris Polak, discussing the handling of long usernames and passwords. The conversation ends with Chris Polak saying "All good!" and "All good!".

```

(base) shermaine@Shermaine-MBP submission % python3 client.py localhost 1234
LOGIN
Enter Username: sup
Enter password: 123
Welcome sup
CREATE
Enter room name you want to create: rom
Successfully created room rom
Waiting for other player...
It's your turn, Please place marker
Type place to make a move, forfeit to end game: PLACE
Col: 0
Row: 0
Validating move: row 0, col 0
Displaying board: 100000000
| X |   |
|   |   |
|   |   |
|---|---|---|
It is the opposing player's turn
sup won due to the opposing player forfeiting
resetting all variables

```

```

% nc -l 1234
LOGIN:hello:123
LOGIN:ACKSTATUS:0
CREATE:rom
CREATE:ACKSTATUS:0
BEGIN:hello:sup
PLACE:0:0
BOARDSTATUS:100000000
BOARDSTATUS:120000000
BOARDSTATUS:120100000
PLACE:0:1
BOARDSTATUS:120120000
PLACE:0:2
BOARDSTATUS:120120100
GAMEEND:120120100:0:hello
| X | O |   |
|   |   |   |
|   |   |   |
|---|---|---|
It is the current player's turn
Type place to make a move, forfeit to end game: PLACE
Col: 0
Row: 1
Validating move: row 1, col 0
Displaying board: 120100000
| X | O |   |
| X |   |   |
|   |   |   |
|---|---|---|
It is the opposing player's turn
Displaying board: 120120000
| X | O |   |
| X | O |   |
|   |   |   |
|---|---|---|
It is the current player's turn
Type place to make a move, forfeit to end game: PLACE
Col: 0
Row: 2
Validating move: row 2, col 0
Displaying board: 120120100
| X | O |   |
| X | O |   |
| X |   |   |
|---|---|---|
It is the opposing player's turn
Congratulations you won
resetting all variables

```

NC client to test server functionality (Game related)

In the first game between hello and SUP, SUP forfeits via CTRL+D, resulting in a default win for hello. P3(testuser) and P4 (super) begin as viewers, with P4 becoming a player in the second game against hello. testuser joins second game as a viewer and attempts to send commands to the server, server ignores messages as viewers cannot interfere with the game.

TERMINAL

```
(base) shermaine@Shermaines-MBP submission % python3 server.py config.json
Server is listening on localhost:6507
Received data from client: LOGIN:hello
o:123

Received data from client: CREATE:rom
Received data from client: PLACE:0:0
Received data from client: LOGIN:SUP:
123
Received data from client: LOGIN:test
user:123
Received data from client: LOGIN:supe
r:123
Received data from client: JOIN:rom:V
IEWER
Received data from client: JOIN:rom:P
LAYER
Received data from client: JOIN:rom:V
IEWER
Received data from client: Client disconnected: Client disconnected
Room 'rom' has been cleaned up and deleted.
Client disconnected and cleaned up
Received data from client: CREATE:rom
Received data from client: JOIN:rom:V
IEWER
Received data from client: JOIN:rom:P
LAYER
Received data from client: FORFEIT
ignoring as its viewer
Received data from client: PLACE:1:1
ignoring as its viewer
```

(base) shermaine@Shermaines-MBP submission % nc localhost 6507
LOGIN:hello:123
LOGIN:ACKSTATUS:0
JOIN:rom:PLAYER
JOIN:ACKSTATUS:0
BEGIN:hello:SUP
BOARDSTATUS:10000000
GAMEEND:10000000:2:hello
CREATE:rom
CREATE:ACKSTATUS:0
BEGIN:hello:super

(base) shermaine@Shermaines-MBP submission % nc localhost 6507
LOGIN:tesuser:123
LOGIN:ACKSTATUS:0
JOIN:rom:VIEWER
JOIN:ACKSTATUS:0
BEGIN:hello:SUP
BOARDSTATUS:10000000
GAMEEND:10000000:2:hello
JOIN:rom:PLAYER
JOIN:ACKSTATUS:0
BEGIN:hello:super

(base) shermaine@Shermaines-MBP submission % nc localhost 6507
LOGIN:super:123
LOGIN:ACKSTATUS:0
JOIN:rom:VIEWER
JOIN:ACKSTATUS:0
INPROGRESS:SUP:hello
JOIN:rom:PLAYER
JOIN:ACKSTATUS:0
BEGIN:hello:super

Server configuration

Invalid length of arguments

```
● (base) shermaine@Shermaines-MBP submission % python3 server.py
Error: Expecting 1 argument: <server config path>.
● (base) shermaine@Shermaines-MBP submission % python3 server.py config.json users.json
Error: Expecting 1 argument: <server config path>.
✖ (base) shermaine@Shermaines-MBP submission % python3 server.py conf.json
Error: conf.json doesn't exist.
```

Invalid Port range

```
1  {
2    "port": 1001,
3    "userDatabase": "users.json"
4 }
```

```
(base) shermaine@Shermaines-MBP submission % python3 server.py config.json
Error: Port is not valid. It should be an integer between 1024 and 65535.
(base) shermaine@Shermaines-MBP submission %
```

Invalid JSON format

A screenshot of the Visual Studio Code interface. The top bar shows tabs for 't_users.json', 'test_config.json', and 'config.json'. The 'config.json' tab is active. The code editor displays the following JSON snippet:

```
1 {  
2   "port": 6507,  
3   "userDatabase": "users.json"  
4 }
```

The word 'users.json' is underlined with a red squiggly line, indicating a syntax error. Below the editor is a status bar with tabs for 'PROBLEMS' (containing 1 error), 'OUTPUT', 'TERMINAL', and 'PORTS'. The terminal panel is open and shows the following command-line session:

```
(base) shermaine@Shermaines-MBP submission % nc -l 1234  
REGISTER:superpiggyng:123  
^C  
(base) shermaine@Shermaines-MBP submission % python3 server.py  
config.json  
Error: config.json is not in a valid JSON format.  
(base) shermaine@Shermaines-MBP submission %
```

Missing keys

A screenshot of the Visual Studio Code interface. The top bar shows tabs for 'server.py M', 'config.py M', and 'config.json M'. The 'config.json' tab is active. The code editor displays the following JSON snippet:

```
1 {  
2 }
```

The code editor highlights the closing brace of the object on line 2 with a red arrow, indicating a syntax error. Below the editor is a status bar with tabs for 'PROBLEMS' (containing 1 error), 'OUTPUT', 'TERMINAL', and 'PORTS'. The terminal panel is open and shows the following command-line session:

```
(base) shermaine@Shermaines-MBP submission % python3 server.py config.json  
Error: config.json missing key(s): port, userDatabase  
(base) shermaine@Shermaines-MBP submission %
```

Missing userDatabase key

A screenshot of the Visual Studio Code interface. The top bar shows tabs for 'server.py M' and 'config.json M'. The 'config.json' tab is active. The code editor displays the following JSON snippet:

```
1 {  
2   "port": 1001  
3 }  
4 }
```

The code editor highlights the closing brace of the object on line 4 with a red arrow, indicating a syntax error. Below the editor is a status bar with tabs for 'PROBLEMS' (containing 1 error), 'OUTPUT', 'TERMINAL', and 'PORTS'. The terminal panel is open and shows the following command-line session:

```
(base) shermaine@Shermaines-MBP submission % python3 server.py config.json  
Error: config.json missing key(s): userDatabase  
(base) shermaine@Shermaines-MBP submission %
```

Missing port key

```
server.py M config.json X
submission > {} config.json > ...
1 > {
2 |   "userDatabase": "users.json"
3 }

(base) shermaine@Shermaines-MBP submission % python3 server.py config.json
Error: config.json missing key(s): port
(base) shermaine@Shermaines-MBP submission %
```

Unknown keys

```
server.py M config.py M config.json X
submission > {} config.json > ...
1 {
2   "port": 6507,
3   "userDatabase": "users.json",
4   "hello" : "testignore"
5 }

(base) shermaine@Shermaines-MBP submission % python3 server.py config.json
Server is listening on localhost:6507
■
```

userDatabase DNE

```
server.py M config.py M config.json X
submission > {} config.json > userDatabase
1 {
2   "port": 6507,
3   "userDatabase": "bla/bla/users.json"
4 }

↑
DNE

(base) shermaine@Shermaines-MBP submission % python3 server.py config.json
Error: bla/bla/users.json doesn't exist.
(base) shermaine@Shermaines-MBP submission %
```

Invalid JSON format

A screenshot of a terminal window titled "zsh - submission". The terminal shows code in a file named "users.json". There are two red arrows pointing to the closing brace on line 37 and the opening brace on line 34, indicating a syntax error. The command "python3 server.py config.json" is run, resulting in the error message: "Error: users.json is not in a valid JSON format."

```
submission > {} users.json > () 8 > password
27     "username": "testuser",
28     "password": "$2b$12$d0lFMLf31e.P2DQ08MJHm.j64RnwWy0jh2d14G6qVLCMECMYYI826"
29   },
30   {
31     "username": "testuser2",
32     "password": "$2b$12$Jw0qFNnYBqqy7IvPckwF2uFfzrISVTfJAY.jshXk6MMpdrWiAV8YC"
33   },
34   {
35     "username": "dummytest",
36     "password": "$2b$12$L.K5vTnafAyiNuYkN1UnUe6DqAjLa502hWFxg5M2MgQa9/VUjpJWC"
37 }
```

```
(base) shermaine@Shermaines-MBP submission % python3 server.py config.json
Error: users.json is not in a valid JSON format.
(base) shermaine@Shermaines-MBP submission %
```

Not in array (missing [])

A screenshot of a terminal window titled "zsh - submission". The terminal shows code in a file named "users.json". A red annotation "missing []" is written over the opening brace on line 3. The command "python3 server.py config.json" is run, resulting in the error message: "Error: users.json is not a JSON array."

```
submission > {} users.json > ...
1
2   {
3     "username": "dummytest",
4     "password": "$2b$12$L.K5vTnafAyiNuYkN1UnUe6DqAjLa502hWFxg5M2MgQa9/VUjpJWC"
5 }
```

```
(base) shermaine@Shermaines-MBP submission % python3 server.py config.json
Error: users.json is not a JSON array.
(base) shermaine@Shermaines-MBP submission %
```

Missing password key

A screenshot of a terminal window titled "zsh - submission". The terminal shows code in a file named "users.json". A red annotation "missing pwd key" is written over the "password" field on line 35. The command "python3 server.py config.json" is run, resulting in the error message: "Error: users.json contains invalid user record formats."

```
submission > {} users.json > () 8 > username
27     "username": "testuser",
28     "password": "$2b$12$d0lFMLf31e.P2DQ08MJHm.j64RnwWy0jh2d14G6qVLCMECMYYI826"
29   },
30   {
31     "username": "testuser2",
32     "password": "$2b$12$Jw0qFNnYBqqy7IvPckwF2uFfzrISVTfJAY.jshXk6MMpdrWiAV8YC"
33   },
34   {
35     "username": "dummytest"
36   }
37 ]
```

```
(base) shermaine@Shermaines-MBP submission % python3 server.py config.json
Error: users.json contains invalid user record formats.
(base) shermaine@Shermaines-MBP submission %
```

Client Launching test

Inactive port

```
(base) shermaine@Shermaines-MBP submission % python3 client.py localhost 6508
Error: cannot connect to server at localhost and 6508.
```

Incorrect number of arguments

```
④ (base) shermaine@Shermaines-MBP submission % python3 client.py localhost
Error: Expecting 2 arguments: <server address> <port>
```

In/Out Game Player Commands

Player inputs anything other than allowed commands, re-prompt.

```
(base) shermaine@Shermaines-MBP submission % python3 client.py localhost 6507
LOGIN
Enter Username: hello
Enter password: 123
Welcome hello
GAME
Unknown command: GAME
SIDEE
Unknown command: SIDEE
CREATE
Entered room name you want to create: room1
Successfully created room room1
Waiting for other player...
It's your turn, Please place marker
Type place to make a move, forfeit to end game: FLIFF
Unknown command: FLIFF
Type place to make a move, forfeit to end game: HOWTO
Unknown command: HOWTO
Type place to make a move, forfeit to end game: HELP
Unknown command: HELP
Type place to make a move, forfeit to end game: 
```

```
○ (base) shermaine@Shermaines-MBP submission % python3 client.py localhost 6507
LOGIN
Enter Username: SUP
Enter password: 123
Welcome SUP
JOIN
Enter room name to join: room1
Join as (Player/Viewer): PLAYER
Successfully joined room room1 as a PLAYER
It is the opposing player's turn
□
```

```
○ (base) shermaine@Shermaines-MBP submission % python3 server.py config.json
Server is listening on localhost:6507
Received data from client: LOGIN:hello:123
Received data from client: LOGIN:SUP:123
Received data from client: JOIN:room1:PLAYER
□
```

Client Quits or Receives Unknown Command from server

```
④ (base) shermaine@Shermaines-MBP submission % python3 client.py localhost 1234
client.py localhost 1234
Welcome None
Unknown message received from server. Exiting...
Connection closed.
```

```
● (base) shermaine@Shermaines-MBP submission % nc -l 1234
LOGIN:ACKSTATUS:0
UNKNOWNCOMMANDFROMSERVERTERMINATE:0
○ (base) shermaine@Shermaines-MBP submission % 
```

```
④ (base) shermaine@Shermaines-MBP submission % python3 client.py localhost 1234
QUIT
Connection closed.
```

```
● (base) shermaine@Shermaines-MBP submission % nc -l 1234
LOGIN:ACKSTATUS:0
UNKNOWNCOMMANDFROMSERVERTERMINATE:0
● (base) shermaine@Shermaines-MBP submission % nc -l 1234
```

Relative Path provided

Added a users.json file in my ~ dir and ran my program to test if my program expands ~ correctly.

```
{ config.json M ×
submission > { config.json > ...
1  <-
2    "port": 6507,
3    "userDatabase": "~/users.json"
4 }
```

```
PROBLEMS OUTPUT TERMINAL PORTS
> ✎ TERMINAL
( base ) shermaine@Shermaines-MBP INFO1112 Assignment % cd submission
( base ) shermaine@Shermaines-MBP submission % python3 server.py config.json
Server is listening on localhost:6507
Received data from client: LOGIN:hello:123
□
```

```
( base ) shermaine@Shermaines-MBP submission % python3 client.py localhost 6507
LOGIN
Enter Username: hello
Enter password: 123
Welcome hello
□
```

Config.json

```
test_server.in {} config.json X
submission > {} config.json > ...
1  {
2      "port": 6507, ← port used
3      "userDatabase": "users.json"
4 }
```

Users.json (all users involved above were in userDatabase)

```
[{"username": "rom", "password": "$2b$12$Kb0vU$BwH0Jz9FEy8Ex0Ajb/b0F.3/kGbYZ/HpjeaJz0CCugLa"}, {"username": "HELLO", "password": "$2b$12$d/v3y4xCinF5Q7FT8EB6NeAzcPNdcBw9CV9dEgp4U7tM47es030N0"}, {"username": "myboi", "password": "$2b$12$SHUWSCIuS.C02fryw.uYcb.Ke0P7h5vRtVmqlERgW0ghoJ/eS.03ZS"}, {"username": "SUP", "password": "$2b$12$.qK8sXg/B0eJDqDA2U6rGeDbmdGWhjeqUSSapAgsDTuRnvjUL7uSS"}, {"username": "hello", "password": "$2b$12$A88Q2/1dYe8En0u0Px1Jp0bipnv8keI.omc.vG6GmggDj00ZgztbG"}, {"username": "super", "password": "$2b$12$HfMi19.zQm5jdUfwLlCTveEt5dMCLGcOsauQCTM.craEXzdz16Uu2"}, {"username": "testuser", "password": "$2b$12$dQ1FMf31e.P2DQ08MJHm.j64RnwWy0jh2d14G6qVLOMECMYYI826"}, {"username": "testuser2", "password": "$2b$12$Jw0qFnNYBqqy7IvPckwF2uFfzrISVTf3Ay-jshXkGMpdrWiAVBYC"}, {"username": "dummytest", "password": "$2b$12$L.K5vTnafAy1NuYkNIUnUe6DqAjLa502hWFxg5M2Mg0a9/VUjpJWC"}]
```

ALL FEATURES TESTED