DSL ASSIGNMENT 8

CODE

server.py

```
import socket # for networking
import pickle # for sending/receiving objects

# import the game
from tic_tac_toe import TicTacToe

HOST = '127.0.0.1' # this address is the "local host"
PORT = 12783       # port to listen on for clients

# set up the server
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((HOST, PORT))
s.listen(5)

# accept a connection from the client
client_socket, client_address = s.accept()
print(f"\nConnnected to {client_address}!")

# set up the game
player_x = TicTacToe("X")

# allow the player to suggest playing again
rematch = True

while rematch == True:
    # a header for an intense tic-tac-toe match!
    print(f"\n\n T I C - T A C - T O E ")

    # the rest is in a loop; if either player has won, it exits
    while player_x.did_win("X") == False and player_x.did_win("O") == False and
player_x.is_draw() == False:
        # draw grid, ask for coordinate
        print(f"\n       Your turn!")
        player_x.draw_grid()
        player_coord = input(f"Enter coordinate: ")
        player_x.edit_square(player_coord)
        # draw the grid again
        player_x.draw_grid()

        # pickle the symbol list and send it
        x_symbol_list = pickle.dumps(player_x.symbol_list)
        client_socket.send(x_symbol_list)

        # if the player won with the last move or it's a draw, exit the loop
        if player_x.did_win("X") == True or player_x.is_draw() == True:
            break

        # wait to receive the symbol list and update it
        print(f"\nWaiting for other player...")
        o_symbol_list = client_socket.recv(1024)
        o_symbol_list = pickle.loads(o_symbol_list)
        player_x.update_symbol_list(o_symbol_list)

    # end game messages
    if player_x.did_win("X") == True:
        print(f"Congrats, you won!")
    elif player_x.is_draw() == True:
        print(f"It's a draw!")
    else:
        print(f"Sorry, the client won.")

    # ask for a rematch
    host_response = input(f"\nRematch? (Y/N): ")
    host_response = host_response.capitalize()
    temp_host_resp = host_response
    client_response = ""

    # pickle response and send it to the client
    host_response = pickle.dumps(host_response)
    client_socket.send(host_response)

    # if the host doesn't want a rematch, we're done here
    if temp_host_resp == "N":
        rematch = False

    # if the host does want a rematch, we ask the client for their opinion
    else:
        # receive client's response
        print(f"Waiting for client response...")
        client_response = client_socket.recv(1024)
        client_response = pickle.loads(client_response)

        # if the client doesn't want a rematch, exit the loop
        if client_response == "N":
            print(f"\nThe client does not want a rematch.")
            rematch = False

        # if both the host and client want a rematch, restart the game
        else:
            player_x.restart()

spacer = input(f"\nThank you for playing!\nPress enter to quit...\n")

client_socket.close()
```

client.py

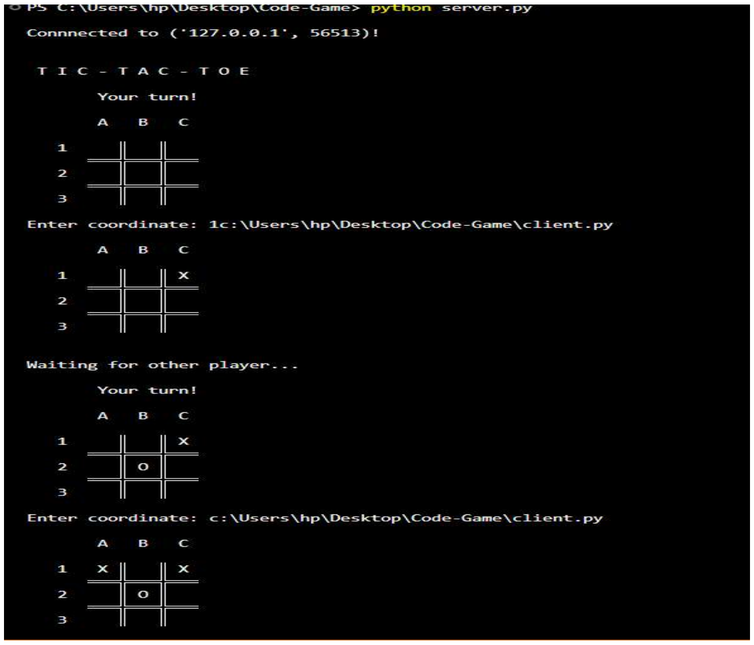
```
import socket # for networking
import pickle # for sending/receiving objects

# import the game
from tic_tac_toe import TicTacToe

HOST = '127.0.0.1'  # the server's IP address
PORT = 12783        # the port we're connecting to

# connect to the host
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))
print(f"\nConnected to {s.getsockname()}!")

# set up the game
player_o = TicTacToe("O")

# allow the player to suggest playing again
rematch = True

while rematch == True:
    # a header for an intense tic-tac-toe match!
    print(f"\n\n T I C - T A C - T O E ")
    # draw the grid
    player_o.draw_grid()

    # host goes first, client receives first
    print(f"\nWaiting for other player...")
    x_symbol_list = s.recv(1024)
    x_symbol_list = pickle.loads(x_symbol_list)
    player_o.update_symbol_list(x_symbol_list)

    # the rest is in a loop; if either player has won, it exits
    while player_o.did_win("O") == False and player_o.did_win("X") == False and
player_o.is_draw() == False:
        # draw grid, ask for coordinate
        print(f"\n       Your turn!")
        player_o.draw_grid()
        player_coord = input(f"Enter coordinate: ")
        player_o.edit_square(player_coord)

        # draw grid again
        player_o.draw_grid()

        # pickle the symbol list and send it
        o_symbol_list = pickle.dumps(player_o.symbol_list)
        s.send(o_symbol_list)

        # if the player won with the last move or it's a draw, exit the loop
        if player_o.did_win("O") == True or player_o.is_draw() == True:
            break

        # wait to receive the symbol list and update it
        print(f"\nWaiting for other player...")
        x_symbol_list = s.recv(1024)
        x_symbol_list = pickle.loads(x_symbol_list)
        player_o.update_symbol_list(x_symbol_list)

    # end game messages
    if player_o.did_win("O") == True:
        print(f"Congrats, you won!")
    elif player_o.is_draw() == True:
        print(f"It's a draw!")
    else:
        print(f"Sorry, the host won.")

    # host is being asked for a rematch, awaiting response
    print(f"\nWaiting for host...")
    host_response = s.recv(1024)
    host_response = pickle.loads(host_response)
    client_response = "N"

    # if the host wants a rematch, then the client is asked
    if host_response == "Y":
        print(f"\nThe host would like a rematch!")
        client_response = input("Rematch? (Y/N): ")
        client_response = client_response.capitalize()
        temp_client_resp = client_response

        # let the host know what the client decided
        client_response = pickle.dumps(client_response)
        s.send(client_response)

        # if the client wants a rematch, restart the game
        if temp_client_resp == "Y":
            player_o.restart()

        # if the client said no, then no rematch
        else:
            rematch = False

    # if the host said no, then no rematch
    else:
        print(f"\nThe host does not want a rematch.")
        rematch = False

spacer = input(f"\nThank you for playing!\nPress enter to quit...\n")

s.close()
```

STEPS:

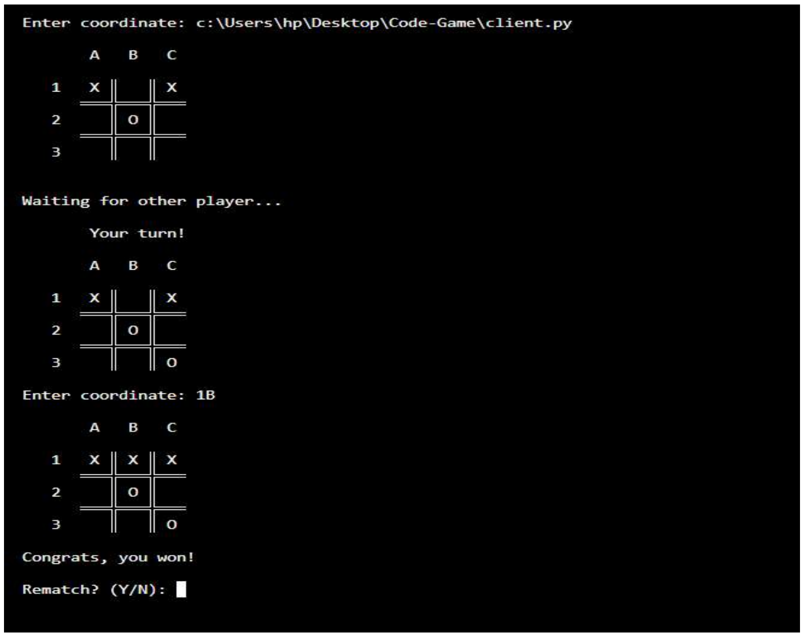
1. Terminal 1
   ```
   python server.py
   ```

2. Terminal 2
   ```
   python client.py
   ```

Terminal 1

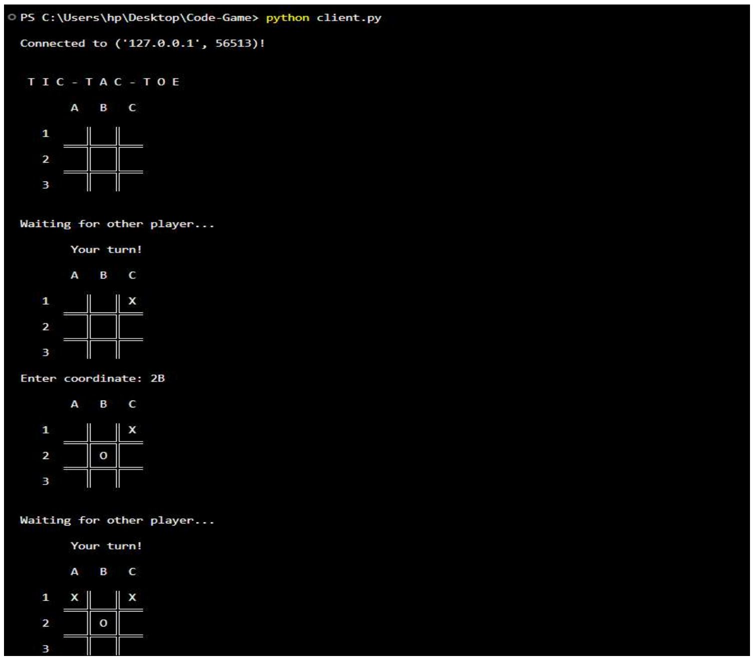


```
PS C:\Users\hp\Desktop\Code-Game> python server.py
 Connected to ('127.0.0.1', 56513)!

  T I C - T A C - T O E
          Your turn!
          A   B   C
   1        ║   ║
         ═══╬═══╬═══
   2        ║   ║
         ═══╬═══╬═══
   3        ║   ║
 Enter coordinate: 1c:\Users\hp\Desktop\Code-Game\client.py
          A   B   C
   1        ║   ║ X
         ═══╬═══╬═══
   2        ║   ║
         ═══╬═══╬═══
   3        ║   ║

 Waiting for other player...
          Your turn!
          A   B   C
   1        ║   ║ X
         ═══╬═══╬═══
   2        ║ O ║
         ═══╬═══╬═══
   3        ║   ║
 Enter coordinate: c:\Users\hp\Desktop\Code-Game\client.py
          A   B   C
   1      X ║   ║ X
         ═══╬═══╬═══
   2        ║ O ║
         ═══╬═══╬═══
   3        ║   ║
```

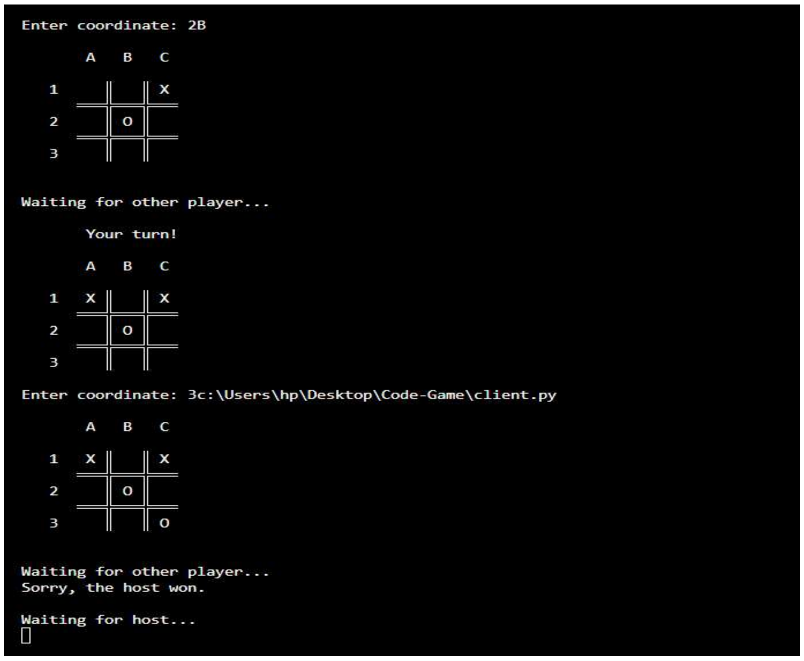


```
Enter coordinate: c:\Users\hp\Desktop\Code-Game\client.py

        A   B   C

  1     X ║   ║ X
       ═══╬═══╬═══
  2       ║ O ║
       ═══╬═══╬═══
  3       ║   ║


Waiting for other player...

        Your turn!

        A   B   C

  1     X ║   ║ X
       ═══╬═══╬═══
  2       ║ O ║
       ═══╬═══╬═══
  3       ║   ║ O

Enter coordinate: 1B

        A   B   C

  1     X ║ X ║ X
       ═══╬═══╬═══
  2       ║ O ║
       ═══╬═══╬═══
  3       ║   ║ O

Congrats, you won!

Rematch? (Y/N):
```

Terminal 2



```
PS C:\Users\hp\Desktop\Code-Game> python client.py
 Connected to ('127.0.0.1', 56513)!

  T I C - T A C - T O E
        A   B   C
   1      ║   ║
       ═══╬═══╬═══
   2      ║   ║
       ═══╬═══╬═══
   3      ║   ║

 Waiting for other player...
        Your turn!
        A   B   C
   1      ║   ║ X
       ═══╬═══╬═══
   2      ║   ║
       ═══╬═══╬═══
   3      ║   ║
 Enter coordinate: 2B
        A   B   C
   1      ║   ║ X
       ═══╬═══╬═══
   2      ║ O ║
       ═══╬═══╬═══
   3      ║   ║

 Waiting for other player...
        Your turn!
        A   B   C
   1    X ║   ║ X
       ═══╬═══╬═══
   2      ║ O ║
       ═══╬═══╬═══
   3      ║   ║
```



```
Enter coordinate: 2B

        A   B   C

  1       ║   ║ X
       ═══╬═══╬═══
  2       ║ O ║
       ═══╬═══╬═══
  3       ║   ║


Waiting for other player...

        Your turn!

        A   B   C

  1     X ║   ║ X
       ═══╬═══╬═══
  2       ║ O ║
       ═══╬═══╬═══
  3       ║   ║

Enter coordinate: 3c:\Users\hp\Desktop\Code-Game\client.py

        A   B   C

  1     X ║   ║ X
       ═══╬═══╬═══
  2       ║ O ║
       ═══╬═══╬═══
  3       ║   ║ O


Waiting for other player...
Sorry, the host won.

Waiting for host...
```