CST 376/476
Michael Ruth
Semester Project
Due: See Description below

The semester project will be a simplified multiuser dungeon (MUD) using the provided code with two components: **MUDServer** & **MUDClient**

- **MUDServer** contains the rooms, maze, players, etc. and models the world (maze) in which players interact.
  - Upon waking, it reads a list of usernames/passwords from a text file.
    - Periodically, the list of usernames/passwords in memory will be written out to the text file for persistence reasons
  - The server will listen for broadcasts from clients (Client connection requests) and upon hearing a client, the server will contact the client with the server's contact information
  - The server manages the users within the world (including clients logging in):
    - If a username already exists, the client may not use them
  - The server handles any requests from clients and deals with them accordingly (see below for list of activities to be supported from the user)
- **MUDClient** connects to the **MUDServer** and deals with the server and the user.
  - This enables your human interaction with the **MUDServer**
  - Must be a GUI with (at least) the following characteristics:
    - Must have a text area for output from the server, a list box for users in the same room as the current user, a text box for input from the client, and a send button to send it…
    - Upon waking up, :
      - The client should broadcast a request for connection message and attempt to connect to the first reply (from the server)
      - After connecting to the server, the client should negotiate a username and set a password. Note: a couple of things can/should happen here
        - The client may send a username/password pair to login (no negotiation, just use this username and this password)
        - The client sends a username request with a username (IE, try meruth)
    - Basic set of events:
      - User clicks on another users name: (requesting/starting a private chat)
        - A request for a private interaction proceeds as follows:
        - The server gives out connection info if the user agrees to the interaction (a dialog box should pop up and ask the user if they want to talk to x)
        - Either side should be able to quit (exit) from the chat by either typing quit or closing the private chat window
      - User interacts with the world by typing various commands and seeing the result from the server in the text area window

- **Some basic functionality**:
  - Change your description information (to whatever you want) such as short description, long description (but your username must not be altered)
  - Move from room to room (move <direction>)
    - Note: when a player moves, all the players in the old room (X is leaving message) and new room (X arrived message) are notified of the move.
  - Talk to everyone in a room (talk/say command)
  - Yell to everyone on server (yell command)
  - Whisper to another user (whisper command) <- talks through server!
  - Look at yourself, the room, the other players, and the exits
    - Notes about the look command:
      - You would look at yourself to see your long description
      - You would look at other players to see their long description
      - You would look at the room to review the room contents:
        - description of room including Room name and description
        - Other players in room (name + short desc)
        - The exits
      - You look at the exits to review exits without refreshing the room

The graded components (and their deadlines) of this project are as follows:

1. **(0%) Team Designation (Due: 3/28):** You need to provide the names of each team members and the team name. Note: I expect duplicates (everyone needs to send this)
2. **(10%) Message Diagrams (Due: 4/4):** Illustration of each message the components will send/receive including the format of the message and its contents
   a. Note: All groups will turn this in and I will review and return a set for the entire class to use for the rest of the project on **4/11**
3. **(10%) Project Design (Due: 4/4):** Doesn't need to be very formal, but you need to layout the classes that will be used and their interaction and responsibilities well before project is due.
   a. I will review these and get back to you to discuss them
4. **(5%) Progress Check (4/18):** You need to **show** me your progress toward completion.
5. **(5%) Progress Check(4/25):** You need to **show** me your progress toward completion.
6. **(70%) Final Deadline (Due: 5/2 – before class):**
   a. **(30%) Server:**
      i. **(20%)** Interaction with clients (basic command implementation (look, talk, move, etc.)
      ii. **(5%)** Private chat interaction
      iii. **(5%)** User management (logins)
   b. **(30%) Client:**
      i. **(5%)** Overall GUI Design
      ii. **(10%)** Interaction with the server
      iii. **(10%)** Interaction with other clients
      iv. **(5%)** Connection protocol using UDP (Clients finding a server and then connecting to it)

c. **(10%)** Conformance to message passing protocol design (IE, reads/writes messages)
    i. This will be tested using servers/clients from other teams!

Special Notes:

For progress checks, someone from your team needs to discuss with me your overall progress and **demonstrate whatever progress you can.** For these checks, your team's grade depends on the actual progress made and your team's likelihood of finishing the project.

For the final deadline, you are required to **demonstrate your application** and test its functionality with other servers in class. On the day of the final, we will meet to perform demonstrations. At least ONE member must be present to demonstrate your client/server. If no one appears for the demonstration, your team (and everyone on the team) will receive a 20 point penalty for failure to demonstrate.

There will be no late projects accepted or graded. If your team fails to turn in your project by the final deadline, I will assume the project was not done and your team will receive a zero for the project.

If the members of your team cannot cooperate, I will assign everyone in the team to either new partners or to me (note: I will NOT work on your server and only work on the client-side of this application).