



C Sc 335 Fall 2010 – Final Project

Overview: A MUD, or Multi-User Dungeon/Dimension/Domain, is a multi-player text environment (The player types commands and the MUD responds with text). This option would involve implementing a functional, networked MUD. They often take the form of an RPG (role-playing game).

Though there are many types and uses of MUDs, all of them share several common elements. A MUD consists of players, rooms, items, and MOBs (MOBile non player characters). Multiple players can connect to the MUD at any given time. They can interact with one another, move about from room to room and collect items that affect them or allow them to do things.

MOBs can also move from room to room, and players may have complex interactions with them. Though most MOBs are hostile towards the player, they do not all need to be. Some MOBs may offer quests, services or games to the players. Such non-hostile MOBs will be referred to a NPCs (Non-Player Characters).

Rooms in a MUD should be thought of as locations rather than an indoor area with walls and a ceiling (A "room" could be a clearing in the woods with a path leading to the north and another to the west, or the middle of a field with paths leading in all directions). Rooms have exits that connect them to other rooms, descriptions, and contents (including items, players and MOBs).

Your Task: You are to implement a fully function MUD Server that multiple players can connect to from different computers via a MUD Client. Your server will run a single game (All players that connect end up in the same MUD).

This project gives you a certain amount of creative leeway. Your MUD could be any type of (appropriate) environment, so long as you have the required elements. Many MUDs take the form of Dungeons and Dragons style Fantasy Adventure games. In such games, players have statistics (health points, strength points, and other things that affect their effectiveness in combat), and engage MOBs in combat in order to gain experience points and advance "levels."

Requirements:

Connections:

1. Multiple players must be able to connect to the MUD at the same time over the Network. To do so, they will launch a MUD Client program which will allow them to connect to the MUD Server running at a different location. Read the Client/Server section for more details.

Players:

2. Players must have a name.
3. Players must have a location (A room they are in).
4. Players must have statistics that are changed via their actions in the MUD. In a traditional RPG, this would be things like hp (health point/hit points), defense, level, etc. For something less traditional, it could be number of questions answered correctly or amount of money won.
5. Players must be able to carry items they find/buy/earn with them as they travel.
6. Players must be able to move between rooms through the use of commands (See Interaction section).
7. Players must be able to **interact with one another**, and be able to see the locations and movements of other players in the same room as them.
8. Players must be able to **interact with MOBs**.

Rooms:

9. The MUD must have at least 30 rooms.
10. A room must have a **description** and a **list of possible exits**, as well as a **list of contents** (items, MOBs, and other players in the room). This must be **viewable** by the player via 'look' and should be displayed by default upon their entrance into the room.
11. You must have at least two rooms with special behaviors, like a locked room which requires a key to be opened.

Items:

12. The MUD must have at least 10 different types of items.
13. All items must have a use in the game. You could have potions to heal damage taken, keys to open doors, special items that affect the player's stats, equipment the player can wear or items needed to finish a quest.

MOBs:

14. The MUD must have at least 10 different types of MOBs.
15. MOBs must be able to move between rooms.
15. MOBs must have **complex behaviors**—these can involve player interactions that take into account player statistics or items a player is carrying, or the contents of the room that the MOB is in, and MOBs should have actions that are triggered periodically, without direct interaction from players. For instance, movement, speech, interacting with other MOBs, etc.

Interaction:

16. When players are in the system, they should be able to send **input** and receive **output** without interference from other players.
17. Players must be able to **interact with the game** with a set of commands that

include **at least** the following (certain commands may be omitted if nonsensical within the context of your MUD):

- A set of movement commands that allows players to navigate through rooms (**north, south, east, west, up, down**, etc)
- **look** (shows description of the room that the player is in, or if an argument is provided, such as an item/player/MOB in the room, it should provide the description of said item/player/MOB). This command gives a 360 degree report of the environment (The player is not assumed to be looking in a specific direction).
- **commands** (lists all the commands useable by a player)
- **ooc** <message> (Out of Character channel—the basic MUD wide chat command—message goes to everyone currently connected)
- **who** (lists all players that are logged in)
- **say** (sends a message to all players in the same room as the player executing the command)
- **tell** <player> <message> (sends a message to only the player targeted)
- **score** (displays the players current status/information)
- **trade** <player> initiates a trade between two players. All transactions must be approved by both players, and will consist of the following sub-commands: **add** <item>, **remove** <item>, **approve**.
- **give** <item> <target> (gives item in your inventory to MOB)
- **get** <item> (gets item from room)
- **get** <item> <target> (get item from MOB - not players)
- **inventory** (lists the items that you are carrying)
- **drop** <item> (drops an item from your inventory to the room)
- **use** <item> (executes the item's default behavior)
- **quit** (allows a player to exit the system—should not shut MUD down)
- **shutdown** (saves the MUD's data and then shuts the system down)
- **emote** <phrase> (prints <player name> <phrase> to System.)

Social commands such as 'giggle' or 'wink' should also be implemented. These commands should have a target and should print <player> <command>s at <target>. System:

18. MUD must be **persistent**. Changes made to players and the MUD must remain even after the MUD is shutdown and restarted.
19. MUD must be able to be **shutdown** from within the game. This function should not be available to all players (ie. Password protected, reserved to specific players, etc).

Consistency:

20. The MUD should be logically consistent. For example, a player should only be able to give items to another player/MOB if they are in the same room. Also, walking north to the next room and then south should take the player back to the original room they were in (Unless the area is designed to randomly connect rooms to create a maze like environment. If so, the area should be clearly marked with a sign before entering it or a warning in the description of the room before entering it).

21. As part of this, you must give descriptions of every exit as well. If a player has just traveled North out of a forest, the description should be something like “To your south you see a dark forest.”

Client/Server:

22. The main portion of the MUD will be written as the MUD Server. All interactions with the client should be sent to the server and should trigger an appropriate response by the server that is sent back to the client. The server should run as a terminal based program on one computer and must accept connections from MUD Client programs that run on other computers over the network.
23. The MUD Client will be a GUI which will allow the user to connect to the server. The MUD Client will connect to the server using a Socket/ServerSocket relationship. It must provide at least three frames (or panels), one of which handles chat (referred to as Chat), one which handles the users’ interactions with the Server and the Server’s responses (referred to as System), and one of which displays some form of geography relative to the MUD environment. This frame, for example, could display an image of the room that the player is in, or a map of all of the rooms that the player has visited, etc.

Note: All commands listed above should work in the System frame (or panel), even if they are a chat command (such as “ooc”). On the other hand, the Chat frame (or panel) should only respond to commands that relate to Chat functions, such as **ooc, who, say, and tell.**

Iteration 1:

+25 Analysis and Design Artifacts

These three artifacts are due on or before Nov 5th in a team meeting

- 1) [AnalysisDocument](#)
- 3) Class Diagram
- 2) Sequence Diagram

User Stories Implemented

Complete the user stories determined during the Thursday team meetings. Minus 10 points for any user story that is not implemented to a max of -50 points.

1. Users can interact with other users on the same server
 2. The state of the player is persistent across multiple sessions and the world doesn't depend on players to exist
 3. Users can execute at least 5 commands more meaningful than chatting
 4. Players can move between rooms
 5. All users are properly notified when relative events occur on the server
-

Iteration 2:

This iteration is your final release. It should implement everything that is required in the specification, and be bug-free. The additional features should be included in Iteration 2.

- +100 Project Specific Requirements
- +25 Robustness, Appearance, Usability
- +50 Additional Features

Additional Features

Implement at least 50 points worth of features from the list below. This is by no means an exhaustive list of features, but merely a list of suggestions for potential features. Some of these are much harder to implement than others, and are worth more accordingly. **Talk to your grader if you have other ideas** to ensure that you receive points for your feature.

- Dropped Connections (5 Points)** – Announce to all when a player’s connection is dropped and save that player’s state.
- Dynamic MUD Editor (15-25 Points)** - An in-game menu system for creating new items, mobs and rooms. This could also use a GUI to simplify the process. This option should not be available to all players.
 - o **Shareable MUDs (5 Points)** – Provide a way for MUDs created with the editor above to be shared.
- Complex Skills/Classes (10 Points)** - Complex skills and classes that allow players to have more sophisticated and a wider variety of interactions with their environment.
- Player Grouping (10 Points)** - if a player joins a group, he or she follows the movement of the leader, and executes certain commands (for instance, attack) if the leader executes them.
- Player Preferences (5-10 Points depending on complexity)** - Player options such as removing oneself from the ooc channel, turning color—if implemented—on and off, etc.
- Abbreviated Commands (5 Point)** - Players able to type abbreviated versions of command names (for instance, ‘l’ for look, ‘sc’ for score, ‘”’ for say, etc). This must be implemented for at least 5 commands.
- Mini-Games (5-15 Points depending on complexity)** – Mini games are games within a game. This could take the form of a card game/dice game or something more complicated. This could be played between players or between players and NPCs.
- Advanced User Interface (5-15 Points depending on complexity)** – Additional Features to the GUI, such as a view of the player's inventory, a health bar, and/or buttons that serve as macros for popular commands, a complex view of the environment, etc.

Server-side commands (5 Points) – Allow an “administrator” to type commands directly into MUD Server which cause responses in the game for all players. Examples include **shutdown**, **restart**, **kick**, **ban**, **banbyIP**, **deleteprofile**, and many more.