**DESIGN PROBLEM: M.U.D.**

## Functional Requirements

Multi-User Dungeon, or M.U.D., is an application with the following features:

1. A character in the game has the following attributes:
   1. Name
   2. Description
   3. Health
   4. Attack
   5. Defense
2. Combat between two characters is fairly simple:
   1. Damage is the attacker's attack stat minus the opposing character's defense stat (minimum of 1), e.g. 10 attack - 5 defense = 5 damage, but 10 attack - 15 defense = 1 damage.
   2. If a character's health is reduced to 0, they are defeated.
3. A human user controls a player character (PC).
   1. Health: 100
   2. Attack: 10
   3. Defense: 0
   4. The name and description are supplied by the user when the PC is created.
   5. An inventory includes a single empty bag with slots to hold up to 6 items. Up to 5 additional bags can be added to the inventory for a total of 6.
      1. The inventory is arranged into a hierarchy:
         1. The inventory
         2. Bags
         3. Individual items
      2. The player may view the following stats at any level in the inventory:
         1. Used/Available space, e.g. 30/36 slots in the inventory, or 4/6 slots in a single bag.
         2. Gold value, e.g. this bag contains items worth 573 gold
      3. The player may choose to destroy an item in their inventory to free up space.
   6. A weapon slot into which one weapon can be equipped. If a weapon is already equipped, it will be swapped into the player's inventory.
   7. An armor slot into which one piece of armor can be equipped. If a piece of armor is already equipped, it will be swapped into the player's inventory.
4. An NPC is controlled by the game. In this version of the game, all NPCs are enemies, but in future versions, some NPCs may not be hostile.
   1. An NPC is either nocturnal or diurnal
   2. Health: 50-150
   3. Attack: 5-15
   4. Defense: 0-10
   5. A description including the stats of the NPC (health, attack, armor).
   6. NPCs do not move but will attack a player that is adjacent to them in any direction (including diagonal).
   7. If an NPC is defeated it may drop an item and/or gold.
5. An item is any object that can be picked up by a player. Items have the following attributes:
   1. Name
   2. Description
   3. Value in gold pieces
6. Some items may be used by a character.
   1. Bags.
      1. If the player is carrying fewer than 6 bags, the new bag is automatically equipped, adding the bag's slots to their inventory capacity.
      2. If the player is carrying 6 bags, new, empty bags will be added to their inventory.
      3. The player may swap a larger bag for an equipped bag. The items are automatically moved from the old bag to the new bag.
   2. Food items restore health.
   3. "Buff" items apply a temporary buff to the player that increases a primary stat for 10 turns. Multiple buff items may be used at the same time, even for the same stat.
   4. Weapons that may be equipped and modify the character's attack stat, e.g. a sword that adds 5 attack.
   5. Armor that may be equipped and modify the character's defense stat, e.g. a tunic that adds 5 defense.
7. A tile represents a 3x3 foot square. A tile may be occupied by:
   1. Nothing (empty)
   2. A trap with an attack value.
      1. Adjacent players have a 50% chance of detecting a trap upon entering an adjacent square.
      2. If a trap is detected, the player has a one-time 50% chance to disarm the trap. Failing to disarm the trap will trigger the trap.
      3. If a player enters a square with an armed trap, the trap is triggered.
      4. When a trap is triggered, it damages the player using the normal mechanics for combat. Once triggered, a trap is disarmed.
   3. A character. A tile that contains a character is considered impassable. If the character is defeated, it is removed from the tile.
   4. A chest containing 1-5 items.
   5. An obstacle, e.g. a boulder, table, etc. A tile that contains an obstacle is impassable.
8. A room is part of the game environment.
   1. A room has the following attributes:
      1. Width in tiles
      2. Height in tiles
      3. Exits on 1 or more walls (north, south, east, west)
      4. A description that includes information about the tiles in the room and any exits. The room may also have its own general description, e.g. "A library with exits on the north and east walls. In the room, you can see a chest, an Angry Ogre (7 health, 10 attack, 5 defense), and a chair."
      5. A string representation suitable for printing to standard output.
9. A map comprises a series of at least two interconnected rooms.
   1. An exit from one room must lead to another room.
   2. One room must be the "start" and one room must be the "goal".
10. The game has a day/night cycle:
    1. The default time between transitions from day to night (and vice versa) is every 10 player turns.
    2. During the day
       1. Diurnal creature stats are increased by 10% (rounded down)
       2. Nocturnal creature stats are reduced by 20% (rounded down)
    3. During the night:
       1. Diurnal creature stats are reduced by 10% (rounded down)
       2. Nocturnal creature stats are increased by 20% (rounded down)
    4. When the day/night cycle changes, the player should be notified via the user interface, and any changes to the NPCs in the same room should be indicated, e.g. "Day turns to night and the Angry Ogre's stats increase by 20%!"
11. During their turn, players may choose from a few basic actions:
    1. Move to an adjacent tile as long as it is not impassable.
    2. Attack one adjacent creature.
    3. Move through an adjacent exit.
    4. Open/loot chests in the same tile.
    5. Disarm traps on adjacent tiles.
    6. At the end of their turn, the player is attacked by any adjacent creatures.
12. The game is over when:
    1. The player reaches the goal and chooses to end the game.
    2. The player is defeated.
13. The user may choose to quit at any time and their progress is saved.

## Non-Functional Requirements

1. The initial version of the application must provide a plain text user interface (PTUI) through which a user may interact with the application using simple, menu-driven commands. There may be requirements for additional user interfaces (e.g. desktop, web, mobile) in future iterations of the product.
2. The user's profile must be stored persistently between restarts of the application. The user should have the option of continuing a previous game in progress or starting a new game.
3. In the first release, it is up to you how to build the rooms/map for the game. It is acceptable to build a map programmatically by hardcoding it provided that it contains all of the elements of the game. You may also choose to invent a file format for saving/loading maps of different sizes and content.
4. ~~As stated in the functional requirements, the time between day to night transitions is 5 minutes. This should be configurable in some way for testing purposes, e.g. the user wants to reduce it to 60 seconds for the purpose of testing combat against nocturnal NPCs.~~