NEA Proposal

# Overview

[TITLE] is a classic-style text adventure game, written in Java. It consists of entering commands to perform actions within the game, while the world is continuously described by the program, who is the narrator. There will be loot and weapons to find, enemies to defeat, and puzzles to solve. There will be enemies of varying difficulty to battle against, and a final boss guarding the finish for the game.

# Features

## Graphical User Interface

There will be a relatively simple GUI for the game, where the narrator will talk to the player, and the player will enter commands. At the top of the screen will be the player’s health, and a box where a dice roll animation will be played when actions relying on chance are performed.

## Login System

When the game first loads, there will be a login screen where the user can register/login to an account. When registering an account, (if it is an available username) the program will hash the password and store it with the username in a local file. When logging in, the program will hash the entered password and check it against the one stored, and if they do not match, the user will not be able to log in.

Due to user details being stored in a file on the computer, it will only be local to that machine, and users will have to make new accounts for separate machines.

## Leaderboard

From the main menu, users will be able to access the leaderboard for the machine they are using. Players’ highest scores will be stored with their username and password, and displayed in a list from greatest to least.

## Inventory System

Throughout the game, there will be items the player can pickup. When they do, the item will be added to their inventory – a space where all their items are stored. At any point, players will be able to check their inventory, and also try to use items from it. Some items will be finite (such as food), meaning the quantity decreases when it is used. However others will be infinite (such as swords) and will have unlimited uses.

When an item is added to the inventory, it will be sorted into an appropriate category (e.g. consumables, weapons), and then sorted alphabetically within their categories based on an insertion sort algorithm.

## Chance Actions

In the game, there will be items and weapons that perform actions based upon chance, similar to the table-top game Dungeons & Dragons.

For example, there could be a sword that has a 25% chance to inflict a critical hit. This would require a roll of at least 15 on a twenty-sided dice. When the player uses the sword, a dice rolling animation would play at the top of the GUI. If the roll was sufficient, the critical hit would be performed.

Other items that would utilise this could be potions, spells, armour, etc. It would also be utilised during attacks from enemies.

## Adaptive Tiles

As the player moves between tiles, the narrator will describe rooms slightly differently depending on what the player has already done there. For example, if the player has already taken an item from a tile, the program will remember and not tell the player about it next time. In addition, if a player has already defeated an enemy, the description will reflect this.

## Adaptive Map

At the top of the GUI, there will be a small map of the world. At the beginning, this map will only show the tile the player starts on, but will expand as they move around to reveal more. The map will be constructed from ASCII pipe characters, with each tile taking up a 3x3 space – the centre being the tile itself, and the edges and corners being part of the surroundings to show walls.

To see if the player has explored a tile, the program will use a HashMap to store the tiles as keys, and booleans for the value. If the value is true, then that tile will be displayed on the map.

### Map Example

A cross with many windows

Description automatically generated with medium confidence

Key:

* Tile where the player can move to
  + ▒
* Wall
  + │ ┌ ─ ┐ ├ ┤ ┼ └ ┘ ┬ ┴
* Door

═ ║