Michael Hecox

Student ID # 00243521

Course # CIS-153-L8

Final Project Initial Plan

I propose for my final project to be the creation of a very simple text adventure game, in which the “game world” is divided up into rooms which can be moved between, has descriptions of each room which can be loaded upon entering them, as well as items which can be picked up and stored in an inventory, left behind to pick up later, or used to open doors or otherwise change the state of some rooms.

the connections between rooms, descriptions of rooms, the rules for what can be interacted with using what, and so on would be loaded from a text document, while a second text document would contain the initial game state, and a third would be used to track the current game state. (potentially the first and second documents could be combined).

When the program starts, the player will be prompted to either start a new game or load a previous one. In either case, the player will be asked to name their save file, however in the event of starting a new game, it will be populated with the “default” game state.

Whenever a player enters a room, or a game state is loaded, a description of that room, and the room’s inventory, shall be loaded and printed

Each room shall have either 4 or 6 directions in which the player can look, with different aspects being visible from each

Items can be freely moved between the room inventory and the player inventory, and what is present and their quantities will be tracked with a dictionary for each inventory

Some data will not be accessible to the player unless they have a specific item in their possession, ex: cardinal direction will not be displayed and cannot be used to navigate unless the player has a compass

When a player attempts to use an item, either on another item or on a room fixture, it will be checked against the possible valid interactions to see what this does, if anything, and the game state will be updated accordingly

When the state of a room is updated beyond simply changing the contents of its inventory, an alternate description and set of valid movement options shall be read

No additional modules besides possibly regular expressions (for parsing) should be necessary for the main implementation

Potential expansion depending on if there is time and if they make sense under scrutiny:

Other entities in the game world, which either walk around randomly every time the player does, or approach the player (this would require random number libraries).

Loading the initial game rules and state from a website rather than a file (would require web stuff, honestly not sure if this is a good idea for reliability reasons)

GUI Presenting the player with a predrawn image of each room (would require art effort or mucking around in 3d software)

GUI and Buttons to trigger some common actions rather than text

Map screen which is filled in as you explore

A diagram of a diagram

Description automatically generated