Madi Binyon

CS172 – 1

Pete Tucker

11 December 2018

Design Specification

I want to create a dungeon crawler RPG where the player in command can spawn in a room, find puzzles to solve, and then once there is nothing left to explore, exit and go to a new room with more puzzles.

In this situation, the system must run a video game, essentially. This means it must have a display for the player to look at, a way for the character to move around, and ways for the player to interact with his/her environment, in this case puzzles in rooms and exits to other rooms. We can assume the player knows how to use WASD key commands and solve simple puzzles.

When creating this game, I want to start from the top, both in coding and in theory, so I started to think of the hierarchy of a dungeon crawler. First, a player will need a dungeon to crawl in. This dungeon will be like a hub of sort for everything else that goes in the game. Inside the dungeon, a player must have rooms to travel through. Inside the rooms, there are walls, floors, exits, puzzles, and of course, the player. I will need classes for the dungeon, the rooms, the player, and the puzzles. The properties of the dungeon will be the most broad, with more specific details happening in the rooms. The dungeon should be able to create rooms (like making a map) and run the game. In order to run the game, the screen must be printed, the player needs to move, the puzzles/exits need to record when/how they are being interacted with, and once the room puzzle is solved, the exit is activated. In order to fit in some extra credit and make the game more diverse, I want to have several different puzzles the player can solve, a few examples being number puzzles and riddles. I want all of the components of the room to be checked based on the location of the player so the game is constantly interacting with itself.