Final Project Design Document

I loved this assignment. As a matter of fact, I think that I am going to keep working on this after I finish this course, simply because I love the concept so much.

When it came to designing this program a lot of things were going through my mind. As always this is when my mind starts to wander into the smaller details before the skeleton of the project was even created. To begin I created a base class called space with a derived class named room. In order to identify each of these rooms, I gave each of them a string with a name. Each of these spaces would have pointers to other spaces. I decided to store all of these spaces into a static array. This would make managing each room easier. Initially, I had all of the rooms point to four additional rooms, but I soon realized that I would have an infinite amount rooms since each one would point to another four rooms and so forth. It took me a while but, when I was playing around with my classes, I decided to create the room "Empty." This would act as my NULL room. Should my program encounter an empty room, I wouldn't display the room. Essentially I was now able to create a room with a single pointer to one additional space.

With that done, I decided that I needed a Person class to wander through the spaces of my creation. The person class was pretty simple as well since at its core is an object that points to a space. And that pointer would only need to point to its current position. This was the skeleton to my final project; a nameless individual that I can place into any of my spaces. Now the initial step was fun in itself, but I needed other Spaces for my rooms to point to besides the empty rooms. It took me a while to think it through, but I decided to create a container class derived from space. I think that given the assignment parameters, the word space was the most appropriate word to use since it doesn't give the same sense of limitations the word room might have. I was having fun making random generic rooms with random generic containers for my

person to interact with, when I decided that my person needed to hold something. I solved this situation when I made an Item class and added a single data member to my person class.

After I finished all of these things to build the skeleton of the game, I decided that it was finally time to look into game themes. The initial thought was to create a dungeon crawling hero that faced many terrible creatures in a deep dark maze. However I found the idea to be rather dull. I was inspired by my eclectic friend Anthony. In one of my conversations with him, he was telling me how he spent a whole week working on his engineering paper and spent most of his time locked up in his room. He had a funny experience afterwards where he smelled some foul odor in his room so he decided to wipe down and clean everything. However to his dismay the odor persisted… It took him a while, but when he pulled out his desk's keyboard tray, he found a box with several old pizzas molding inside. After laughing with him, I knew that this would be the perfect theme.

With a theme in mind, my thought process was streamlined and I had a better sense of where my project needed to go. First things first is I needed to be able to place my person in a room and give them the ability to walk through each room. This was done by making each pointer correspond to a different letter in the WASD gaming control scheme. I think that this would make things easier. The second thing that I did was to create a generic space for my person to wander through. At first it was shaped as a horizontal "I" with the slashes on the top and bottom. I thought that this wouldn't prove to be interesting so I made a stair class derived from the space class. The goal behind the stair class is to force the person from one end of the house to anther portion. Similar to a gate, once the player crosses through, there is no turning back. The last thing that I did was change the Person's ability to hold a singular item into a vector. While I was coding, I had also changed the room structure into a dynamic array of

pointers, but recreating another one for the person seem like such a hassle that I decided to go the easy route and use a vector. The last thing that I needed to do was create a Front Door class. This space would do an item validation check to see if the player can exit/finish the game. With all of that said I was essentially done with the core mechanics of the game.

When it came to the special function, I had to do quite a bit of thinking to figure out what unique thing each space could do. For the room, I made it so that whenever it was activate, the player would gain additional time to wander around. The opposite is true when then player is within a container space. In order to stop the user from abusing the special function in the room or while in a container, I changed it so that the special function would only appear as an option at certain times. For the last two classes, the special function would help facilitate the forceful change in position on the stairs and the item validation for the front door.

One thing that I had to change before the end of the program was the hard coding of the room descriptions. I found that it would take too long for me to edit each room's descriptions as I looked through the program's code. As a result I had to implement a file stream to read from accompanying text files. This definitely cleaned up a lot of codes, but it did require a lot more files.

In terms of bugs and issues, I had a lot of problems with the arrangement of the rooms. Initially used a constructor to assign all of the directional rooms, but this cause a lot of trouble when it came to coding each room since it didn't seem to properly set the rooms in the correct location. I definitely need to investigate into how I can make it more efficient. Secondly when I ran through the program several times, I soon realized that I was not clearing my vector when the game resets. This meant that if the player were to lose a round/ or simply replay the game, the person would have all of the previously obtained items. This was a bug that I almost misses.

As stated in the first line. I really had a lot of fun coding this assignment and I feel that a lot of other students might feel the same way. It required a lot of planning but I think that it was well worth it.