Robert Newton
12/06/16
Reflection

Cs 162

Design Plan:

My program will be a haunted mansion horror/ sci-fi program, it is built around a large mansion that the user must navigate through to get to the “Boss room”. The user wakes up in this room, I got the inspiration when I had the movie saw on late one night. I thought, man how crazy would it be to wake up in a random house, and have no idea what was going on, yet I was required to move around and find my way out. The only difference is I wanted to give the user some help and hints along the way. The user will only have about 50 turns in this program to get it right before the house “engulfs” them. The user must undergo a series of tests to win the game, the user must solve a couple of key problems, collect “key” items, and show an act of kindness.

There is a total of seventeen rooms the user has the option to go through, the user will have the ability to go four directions, each direction the user will be faced with a menu of options, check the compass, move a certain direction, check the current room, display the stability level, and check the inventory. The inventory will store the users picked up items in a bandolier. I chose bandolier because I am a fan of gaming and I think it sounds cooler than backpack or fanny pack.

Another thing I wanted to do was have a rating for the player’s stability, this is what will eventually cause the player to lose their mind. As long as the player is stable the game can continue, otherwise, that is when you become “unstable” and lose your mind. This game will play into the mental mindset as opposed to physical, this will make it more like a horror movie akin to someone losing their mind. Its important to note that the user can trace and check their steps along the way in case they missed an important item or interaction.

Robert Newton
12/06/16
Reflection

If the user is not able to make it through the interactions, say for example, if the player messes up on the interaction with the equation, or does not help the man, they will have an opportunity. This is to ensure the user has a chance to get the keys to open the doors. The doors need to work as a gateway of sorts for the user, that way each room presents its self in a new fashion.

Testing Plan:

The strategy used to test this program will be simple, implement the design incrementally. This will start with the easiest to create, to get the ball rolling, as well as the most independent that way testing them separately when they are completed is easier. I will need to make sure the file structure is setup in a way that I can quickly and easily move throughout each item in a “flowy chart” type of way. This will make implementation easier and more reliable as I can make sure the links are completely and there is no running connections where a certain object or function is not pointing to anything.

Once the functions have been tested, I will continue to build them depending on which ones depend on those that have already been tested. Each one will be unit tested, and finally I will build the rest of the functions that rely on other functions and classes that may be a “foundation”. As the incremental testing goes, I will ensure that I add the more complicated bits later one as more testing gets finished until everything is satisfied. The testing will be complicated to say the least because the game will have to be played. I will play it as a “featureless” game until I simply cannot anymore and must play the full game. This will require an insane amount of print statements, which will be useful, because this is a text based game. The menu is the thing I will need to test later, because the user has multiple options, I need to make sure they work will all the things they depend on.

I will probably ask my wife to play the game a thousand times, luckily I have a comfortable couch in case she gets mad at me for interrupting her with her nursing school. Once submitted, unzipping the file and running the program should suffice for the assignment.

Robert Newton
12/06/16
Reflection

Test Results:

For the most part I followed my testing plan. I will mention, the unit testing with simple functions and the functions they depended on was just fine. Once I worked on the space class as well as the UsrInv class, things got a bit sketchy. Mainly things were not lining up anymore for some reason, and at one point my terminal just simply stopped counting errors because there were too many. That eing said, I had to change one function name as well as a variable and things were just fine. I think this occurred because of exhaustion? It was simply just a mistake of having a large program with several files inside. My menu was off here and there as well but that of course was fixed with minor tweaks, it was a small off-by one error with the check compass and go a certain direction, once that was fixed, everything was find.

Since I had the help of family to play this game, I got some great feedback on how to make the game more fun, as well as some bugs here and there but overall this was extremely fun. I was worried I didn't learn enough but once I got through with this assignment, I am so confident going forward.