Requirements and Specifications Dungeon Master Team

* We will begin to approach the problem by creating a basic framework for the game. The framework has to have a way for the character to move through the dungeon and a way to start the game. On top of that, we can build everything.
* Some challenges that we anticipate include linking rooms inside our dungeon together. We intend to use a door object to link rooms, but it is unclear how we will get this to work. Another big challenge is going to be user input. We need to find a way to make the game user-friendly yet still have plenty of options to choose from in the different situations that will be present. A third major challenge to our dungeon game is the organization of items. We are unsure how we are going to organize items: whether we are going to get them from a file or whether we are going to have a hardcoded list of items. We are also still deciding whether to have item subclasses. We know that there will be different items, but we are unsure quite how to organize them yet.

Project Management Plan

* Because of the nature of the project, all of us might be working on the same functions. This will be especially true for the dungeon class, which all of our functions will link into and take commands.
* Derek will handle the movement through the dungeon. This means that Derek will make a grid-based room system with different tiles for various things such as items and monsters. He will also handle movement between rooms and inside rooms. This means that Derek will mostly deal with interaction between objects, so he will probably be adding methods to most classes.
* Cyruz will handle the stats and AI for monsters as well as the stats for characters. He will give priorities to monsters (best option given the context). Cyruz will also deal with how the monsters and characters will use items. Cyruz will do the attack, defend, flee, and use item commands of monsters.
* Sam will handle the item system. The item object connects with almost every other object in the game, which means that Sam will probably write 1-2 functions in many classes. Sam will determine whether or not items will have subclasses and what the subclasses are. Lastly, Sam will determine how the character and monsters can use items.

Timeline:

1. Have a working prototype by April 30 that we will be able to test and add minor/major fixes. The prototype needs to be similar to the basic framework that is at the top of this document.
2. Have all systems working by May 5 and have all of them connected to each other.
3. Derek needs to have a basic movement structure by April 28-30.
4. Cyruz needs to have a basic character structure by April 28-30.
5. Sam needs to have a basic item structure by April 30 (Because items, while important, are not vital for the program to have a working prototype).
6. By May 7 the program should be mostly done and we should be starting on the PPT presentation.