1 Assignment

This week we are extending last weeks assignment to make a more complex game. We are adding monsters to the game. In completing this assignment you will be working in the same github repository as you did last week. To complete this assignment you will create a branch in your repo and work on the assignment in the new branch. This page has some instructions that may help.

`https://help.github.com/articles/creating-and-deleting-branches-within-your-repository/`

One of the key goals in this assignment is to help you to understand the experience of extending code that was written for one purpose to a new or larger purpose that was not explicitly planned for in the original design.

2 JSON Schema

The JSON schema has been modified. Make sure to note these modifications carefully and make the corresponding changes in your code.

```
Direction := {
  directionName      : String
  room               : String
  enabled            : String
  validKeyNames      : String []
}

Item := {
  name               : String
}

Room := {
  name               : String
  description        : String
  directions         : Direction []
  items              : Item []
}

Player := {
  items            : Item []
}

Layout := {
  startingRoom     : String
  endingRoom       : String
  rooms            : Room []
  player           : Player
}
```

The enabled field of `Direction` says if the direction can be used or not. If the string is “true” it can be used and if it is “false” it can not be used. To enable a direction that is disabled the player must use the command of the form “`use` *Item* `with` *Direction*”. This will check to see if the player has the item and if that item name matches a name in the `validKeyName` array of the direction. If it does it should enable the direction.

The `Items` field of `Room` is the items that start in rooms. They may be picked up by players using the command of the form “`pickup` *item*”.

Instead of providing a sample JSON file for you to develop with, you should write your own JSON file that follows the format above.

3 Game Extension

You must also extend your game engine in an substantial and interesting way. This can add to the JSON so that you have a different JSON format than the standard. Examples of reasonable extensions include the following:

- Add monsters that can be fought.
- Add items that allow teleportation.
- Add ability to play with more than one player.
- Add simple puzzles or challenges to pickup keys.
- Add total moves or time to complete the game and leaderboard.
- Add hidden rooms/directions.

This extension will be constitute 30% of your total work grade for this assignment so choose an extension of sufficient complexity and technical difficulty. Extensions that are not substantial enough will receive a reduced grade. If you are unsure if an extension is sufficient e-mail cs126fa18@gmail.com with your extension idea.

4 Optional Features

This time around, the optional features are up to you. There are lots of ways that you can expand and enhance the game that we have specified so far. Be creative. Once again, the more effort that you put into optional features, the more extra credit it is worth.

Design and Style: As always, use the best design and coding style that you are familiar with. In particular, for this assignment we want you to pay particular attention to your *object decomposition*, that is how you break down the functionality of the assignment into classes and functions (static or non-static) of those classes. Be sure that code related to a particular object is in that class!

Process: You should continue to work on your code so as to implement small bits of functionality, testing them, getting them to work, and committing that piece of work. We expect to see a series of commits in your github with good explanatory commit messages.