CSCI 2120
Homework 3 : Serialization

Introduction

For this assignment you will, starting from an implementation of an adventure game, add the ability to save or load a game via object serialization.

Procedure

1) You are provided an implementation of a text-based adventure game. Currently the game responds to directional movements that allow you to move your adventurer around a maze (that is read from a text file). Add the ability to save (and load) the game using object serialization. Write the serialization of the file out to a default filename SavedGame.dat. Make sure that you take care of any exceptions that may occur when reading or writing to a file (for example: your program should not crash if a save file does not exist!). Errors should be written to the standard error output stream.

You should begin by diving into the provided implementation and getting a feel for how the game works. Start in `Game.java` and work your way down. Once you've compiled the code, you can run the game by typing in your shell:

```
cat d1.txt - | java AdventureGame
```

Type w, s, a, or d to move around. You can optionally pass in the name of a `systemSetting` String (see `ConfigFactory.java` for more details) to get a fully-colored version of the game. To add the ability to save and load, you should add a new `Action` enum type corresponding to `SAVE` and `LOAD`. Then follow the logic of the application to determine where best to add your serialization features. The `Dungeon` class should be updated to implement `Serializable` and your `Game` class should be able to update its `dungeon` variable when a new game is loaded.

It is generally good program design to separate the user interface (UI) from the model (the classes that "do the computation") as much as possible. You'll note that this design rule has been implemented in the starter code. The thing you should be serializing here is whatever the object / objects are that contain a representation of the "state" of the program. The UI, if designed properly, should not have state that effects the logic of the game. Given that, what should you serialize that contains EVERYTHING relating to the state of the game?? Once this choice is made, the serialization should occur somewhere OUTSIDE that object (i.e. if you are going carry a box of toys to your garage for storage, you don't do it from INSIDE the box of toys, right?)

BONUS) – When the user chooses to save or load, allow the user to specify the filename of the save file. **Exception handling is also expected, and this code should enforce that the filename ends in ".dat"**. (**15 points)**

Submission

You will add, commit, and push your program to Gitlab. Label your homework folder "HW3" in your repository.

Grading

Your code should provide fully functional object serialization and the game should "play" cleanly. Any text you output should not corrupt the display.