CS 145: Homework 4

Interactive Fiction

1 Description

Your primary goal in this homework is to explore the creation of your own objects. You will do this in the context of writing a "room" for a work of interactive fiction, as demonstrated in lecture. After the homework deadline, you instructor will stitch all the rooms together and release the collection on Piazza.

You will write at least four classes for this homework: one named `Room` and at least two others, described below. Place them all in the `hw4.username` package, where username is replaced by your actual UWEC username. **All classes and methods must have Javadoc documentation**.

2 Requirements

You've probably noticed that following instructions is a pretty big deal in this course and computing in general. We're loosening the reins now so that you can design your own objects. The requirements are:

- Your `Room` class must have a method named `enter` that takes a `Player` argument and returns a `boolean`. Upon entering, a short description of the room and its contents is printed, after which the player interacts with any objects and characters in the room through a command-response loop. The player's objective is to exit with her pulse intact. How she can successfully do so is up to you, but the general convention is to require the player to solve some sort of puzzle. If she does so and lives, `true` is returned. If she meets an early end, `false` is returned.

- Use the simple `Player` class we wrote in lecture, and put it in the `hw4` package. We will test your code with this version—so modify it at your peril.

- You must write two classes representing objects or characters that the player can interact with in the room. These objects may unlock a door, break down a barricade, pacify an opponent, imbue the player with superpowers, etc. The objects themselves should have at least one method that changes at least one instance variable. (For example, a can of gasoline may be less full when poured out.) Each should have a constructor.

- You objects will be graded on design. Use methods intelligently, as if they were cues to the objects on which they're invoked. Delegate responsibility to the class to simplify the code in `Room`.

- The command-response loop for your `Room` must be able to parse input without crashing. Handle common actions like "look" or "take," as well as verbs pertaining to your objects and characters. Other input should be gracefully ignored, with a response like "I don't understand that."

- You are welcome to write as many objects as you desire. If you want to write more rooms, name them `Subroom0`, `Subroom1`, etc., and have your `Room.enter` method visit each in turn.

3 Submission

See `hw1` for details on the other expectations upon which you will be graded and submission instructions. The SpecChecker will not test that you have appropriately named classes, but it will collect all your files in the `hw4.username` package for you to submit. Parts of the SpecChecker have changed for this homework, so it is necessary that you remove SpecCheckers for previous assignments from the Build Path. The old code may run instead of the new code, leading to exceptions.