CSCI 345
Assignment 5
Due June 10, 2013

Introduction

For this assignment you are to add features to the adventure engine from Assignment 4.

Problem Statement

For this version of your adventure engine, you are to add the following three features:

1. Add brief descriptions to Rooms.
2. Add priority and validity to Actions.
3. Add properties to Players, Rooms, and GameItems.

In addition, I want you to update the UML diagram you did in Assignment 4 to reflect the class structure you have implemented.

New Files

I am providing you with the following new files:

- ValidMethod.java – this is an interface similar to ActionMethod that is used for the validity methods for Actions.
- CloakHardCoded.java – a new hard coded game description for the "Cloak of Darkness" game. This replaces the old HardCodedGame.java.
- Game.java – a new version of the Game class that build the new CloakHardCoded game.

Add Brief Descriptions to Rooms

For this change you will need to add a new String attribute to Rooms that is a brief description. The current (long) description will still be available as description. In order to do this you will need to:

1. Add the following method to the interface IRoom:

```
String getBriefDescription();
```

2. Modify your Room class to implement the new method in IRoom.
3. Modify the makeRoom method in IBuilder to include the brief description:

```
IRoom makeRoom(String name, String desc, String brief);
```

4. Modify your Builder and Room classes to construct Rooms with brief descriptions.

Add Priority and Validity to Actions

The purpose of this change is to allow multiple actions to be associated with the same command; the correct action being chosen based on the current situation. In order to make this change you will need to do the following:

1. Modify the IBuilder interface as follows:

   ```
   void makeAction(IVocabTerm vocab1, IVocabTerm vocab2,
                   int priority, ValidMethod valid,
                   ActionMethod action);
   ```

   This change adds the two additional attributes to Actions, a priority and a valid method.

2. Modify your command interpreter so that the action for a command is chosen action that satisfies the following three conditions:

   1. The vocabs for the action match the words entered by the user. This is the same as your old command interpreter.
   2. There is no valid method for the action or the valid method for the action returns true. (Note: make Action may be called with a null valid method. This is saying that if there is no valid method, the action is considered as valid in all situations.)
   3. The action chosen has the highest priority among all actions that satisfy conditions 1 and 2.

   You can accomplish all of this in a single loop by making decisions about whether the action is usable in the order given. You are guaranteed that no more than one action will be chosen. If there are multiple actions for a given command satisfying conditions 1 and 2, there will be exactly one with the highest priority.

Add Properties

You will need to add properties to the Room, Player, and GameItem classes. Properties are associated with individual objects (no static!). Properties are named by Strings and have a String value.

1. You will need to add the following methods to IRoom, IPlayer, and IGameItem:

   - `String getProp(String propname);`
   - `void setProp(String propname, String value);`

   The method getProp returns the value of the property with name propname. If there is no property with that name, getProp can return null or throw an exception. (CloakHardCoded initializes all properties it uses.)

   The method setProp sets the value of the property with name propname to the value value. If the property does not exist, it should be added to the properties for the object. If it already exists, the new value overwrites the old value.

2. In addition to other properties of a GameItem, the three descriptions associated with a GameItem must also be accessible as properties. For the property names “inventoryDesc”, “hereIsDesc”, and “longDesc” are associated with the inventory description, here is description, and long description for GameItems. So, for example, the method call `item.getProp(“inventoryDesc”)` will retrieve the inventory

description and the method call `item.setProp("inventoryDesc", "a lit stick of dynamite")` should result in a player panic (:-)).

Example

Here's some sample output from my version of the program. (I've provided my version of the program as Assign5.jar. You can run it using `java -jar Assign5.jar`.) I will leave it to you to figure out how to win the game.

```
The Cloak of Darness (Version 0.1)

Hurrying through the rain swept November night, you're glad to see the
bright lights of the Opera House. It's surprising that there aren't more
people about; but, what do you expect in a CS assignment ...?

You are standing in a spacious hall, splendidly decorated in red and
gold, with glittering chandeliers overhead. The entrance from the street
is to the north, and there are doorways south and west.
? inventory
You are carrying: a velvet cloak (worn).
? go south
It is dark in here!
? examine cloak
It's dark in here.
? look east
I have a better idea: let's get out of here before you disturb anything.
? look west
I have a better idea: let's get out of here before you disturb anything.
? go out
You are in the hall.
? examine cloak
It is a handsome cloak, of velvet trimmed with satin, and slightly
spattered with raindrops. Its blackness is so deep that it almost seems
to suck light from the room.
? drop cloak
I don't know how to "drop cloak".
? go west
The walls of this small room were clearly once lined with hooks, though
now only one remains. The exit is a door to the east. There is a brass
hook here.
? examine hook
It's just a small brass hook, screwed into the wall.
? hang cloak
You hang the cloak on the hook.
? inventory
You are empty-handed.
? go out
You are in the hall.
? go south
You are in the bar. There seems to be some sort of message scrawled in
the sawdust on the floor.
? look around
The bar, much rougher than you'd have guessed after the opulence of the
```

```
foyer to the north, is completely empty. There seems to be some sort of
message scrawled in the sawdust on the floor.
? verbose yes
? go out
You are standing in a spacious hall, splendidly decorated in red and
gold, with glittering chadeliers overhead. The entrance from the street
is to the north, and there are doorways south and west.
? go south
The bar, much rougher than you'd have guessed after the opulence of the
foyer to the north, is completely empty. There seems to be some sort of
message scrawled in the sawdust on the floor.
? verbose no
? go out
You are in the hall.
? go south
You are in the bar. There seems to be some sort of message scrawled in
the sawdust on the floor.
? read message
The message has been carelessly trampled, making it difficult to read.
You can just distinguish the words...

YOU HAVE LOST !!!
```

What to Submit

This assignment is due by midnight, June 10.

You are to submit four things:

1. The source for your solution. If you are using Eclipse, you can zip the contents of the `src` directory from your project.
2. A text file showing the results of your test(s) of the version of the game you are submitting.
3. An updated version of your UML class diagram from Assignment 4.
4. A UML sequence diagram showing the sequence of actions that occurs in response to a user command like “go south”. You should assume that the command is valid.

Make a zip file with all four items and submit that zip file on moodle.

Grading

To Be Supplied.