Stage 3: Items

Items are another thing that will make your world interesting. There are many different types of items that will be added to your world:

Play Items: Items that the player must interact with in order to achieve some goal during the game. Most of you have identified these already in your design document.

Bonus Items: Items that exist for the player to interact with to achieve some side goal. None of these items are necessary to win the game but they make the world a more interesting place for the person playing your game.

Immovable Items: Items that a player an interact with but cannot take with him/her. You probably haven't given these much thought. Having immovable items in a Room makes the world more interesting. Examples of immovable item include a couch or a bet, a desk with drawers, a picture on a wall, basically anything that an appear in a Room. Your world is rich with items even if you haven't listed them all in your design document, and most of them are immovable items.

In order to add items to your game, do the following:

1. Implement an Item class. Items start by having a unique name, a description, and a weight. Items will be referred to by their name.
2. Implement a ComplexItem class. A complex item is one that contains other items. The ComplexItem class should extend the Item class adding a field and methods that allow it to hold other items (everybody is required to have complex items). In addition to normal methods (getters, setters, toString, etc), your complex item class needs the following methods:
   - **public void** addItem(Item item) which adds an item to the complex item.
   - **public** String [] getItemNames() which returns an array of names of the items that are contained in the complex item.
   - **public** Item removeItem(String name) which removes the named item from the complex item.
3. Modify your game so that a Room can store a collection of items.
4. Add a method, addItem, to the Room class to add an item to this room. This method only needs a single parameter.
5. Add items to your world by adding a createItems method in your World class and calling it in the constructor. Items are added to the world by placing them in the room in which they belong. Items that are contained inside a complex item can be placed directly in the complex item. You must add all of the items listed in your design document.
6. When a player enters a place, information about the items in the place should be displayed. Before you implement this, consider the following questions:
   1. How much information about an item present in a place should be included?

   2. What class should produce the string describing the list of items in a room?
   3. What class should produce the string describing an item in detail?
   4. Which class should print it?

   If the answers to these questions make you feel like you should change your implementation, go ahead and make the changes (be sure that any changes you make in your code are reflected in your Design Document).

7. Test to make sure that everything is working as expected.

8. Add an "*examine* `item name`" command which can be used to get a complete description of the named item. When a complex item is examined, the items that are contained inside should be listed (unless the complex item cannot be opened).

9. Add an "*unpack* `item name`" command that will remove all of the items inside a complex item and place them in the current room.

10. Add a "*pack* `item` *into* `complex item`" command that will move the specified item into the specified complex item.

11. Modify the `Door` class to store an item that will store a "key" that is used to unlock that specific door. We will be adding using a "key" to unlock a particular door in Stage 4.

12. As usual, continue to improve your game by working on the room and item descriptions.

In order to continue you must:

- Add all the items listed in your design document to the appropriate rooms in your game.
- Add *examine*, *pack*, and *unpack* commands.
- Update your design document to to include other items. For each room in your world, make a list of items in that room. For PLAY and BONUS items, you should include item names, item descriptions, and their purpose.
- Update your design document to include any additional commands you may have added during this stage.

Penalties (will not prevent you from moving on, but will lower your overall score):

- Not having a rich environment by placing lots of items in your world.

When you are ready upload your project archive (from Github) to Moodle for review. Your submission will be reviewed within 48 hours (often sooner). The review process will ensure that you have completed this stage before allowing you to progress to the next stage.

Deadline: Complete this by the assigned deadline (see Moodle).