

Homework 2 CS3330 Fall 2014

Due Thursday, October 16th by 5:00pm

Objectives:

- Use ArrayLists
- Implement Composition and Aggregation
- Improve Java Skills

Purpose:

Homework 2 is the next foundation in our homework project! By the end of the semester, you will have created a complex game that has RPG similarities. The component of the game you are making in this homework is creating an Avatar interacting with his surroundings, allowing for decision to be made by the game player. The program is called RoomEscape, where you will look for weapons to add to your personal bag. Once you defeated all the levels and beasts, you win the game.

HW2 Download Material:

HW2.zip - Contains the UML class diagram, given source code, and the data files for HW2.

Note: The UML class diagram doesn't show any basic getters (public access) and setters (private access), make sure to add those to all your classes except **RoomEscape**.

Submission Information:

cs_submit CS3330_HW-*<LabLetter>* HW2 *<yourpawprint>*.cs3330.hw2.zip

lab**l**etter:

Section **A**, Tuesday 8:00-8:50am

Section **B**, Tuesday 2:00-2:50pm

Section **C**, Monday 11:00-11:50am

Section **D**, Monday 3:00-3:50pm

For reference on how to zip a eclipse project locally, look at LAB 3 Lecture and 'Uploading with Filezilla' Document for submission uploading to babbage.

[Bag.java](#)

Bag () - Calls the **initBag** method.

void initBag() - Assigns the **items** attribute to a new ArrayList of type **Item** and assigns the **weight** attribute to zero.

Item getItem(name : String) - If there is an item in the **items** ArrayList has the same name as the passed parameter **name**, return that item; else, return null.

ArrayList<Item> getItems() - Returns the **items** ArrayList.

int getSize() - Returns the current size of the **items** ArrayList.

boolean addItem(item : Item)

Checks if the total of the current **weight** attribute and the passed item parameter's **weight** is less than or equal to 20. If less, adds the passed item parameter to **items**, adds the item's weight to the attribute **weight**, and returns true. If not, return false.

boolean dropItem(item : Item)

Attempts to remove the passed-in **item** from the **items** ArrayList. If successful, decrease the **weight** attribute by the removed item's weight and return true. If not successful, return false.

[Beast.java](#)

Beast(name : String , hp : int)

Calls the setters **setName** using the passed parameter **name**, and **createHealth** using the passed parameter **hp**.

void setName(name : String) - Assigns the attribute **name** to the passed parameter **name**.

void createHealth(hp : int) - Assigns the **health** attribute to a newly created **Health** object using the passed-in **hp**.

String getName() - Returns the **name** attribute.

int getHealth() - Returns the result of calling the **getHealthPoints** method on the **health** attribute.

boolean injured(item : Item)

Checks if the passed **item** parameter is not null. If not null, call the **hit** method on the **health** attribute with the parameter as the value returned from calling the **getValue** method on the passed **item** parameter; also return true. If null, return false instead.

boolean isLiving() - Returns the result of calling the **getAlive** method on the **health** attribute.

[CommandProcessor.java](#)

CommandProcessor() - Calls the setter **setGameCommands**.

void setGameCommands()

Instantiates a new **GameDataReader** object and calls the **getGameCommands** method on the newly instantiated **GameDataReader** object using the following String "GameData/Commands.txt", and assigns the returning ArrayList of type String to the attribute **gameCommands**.

ArrayList<String> getCommands() - Returns the entire **gameCommands** ArrayList.

boolean validateUserCommand(command : String)

Checks if the passed-in **command** is in **gameCommands**. Returns true if command is found; else, returns false.

[GameDataReader.java](#)

ArrayList<String> getGameCommands(filePath : String)

Instantiates an ArrayList of type **String**. Reads and parses the content of the passed **filePath** parameter into new String objects that are added to the already created ArrayList of type **String**. After reading the entire file, return the populated ArrayList of type **String**.

ArrayList<Item> getGameItems(filePath : String)

Instantiates an ArrayList of type **Item**. Reads and parses the content of the passed **filePath** parameter into new Item objects that are added to the already created ArrayList of type **Item**. After reading the entire file, return the populated ArrayList of type **Item**.

ArrayList<Beast> getGameBeasts(filePath : String)

Instantiates an ArrayList of type **Beast**. Reads and parses the content of the passed **filePath** parameter into new **Beast** objects that are added to the already created ArrayList of type **Beast**. After reading the entire file, return the populated ArrayList of type **Beast**.

[Health.java](#)

Health() - Calls the setters **setHealth** using 100 and **setAlive** using true.

Health(hp : int) - Calls the setters **setHealth** using the passed parameter **hp** and **setAlive** using true.

void setHealth(hp : int)

Checks if the passed parameter **hp** is greater than or equal to 1; if so, assigns the attribute **healthPoints** to the passed parameter **hp**, else call **setAlive** using false.

void setAlive(alive : boolean) - Assigns the attribute **alive** to the passed parameter **alive**.

int getHealthPoints() - Returns the attribute **healthPoints**.

boolean getAlive() - Returns the attribute **alive**.

void hit(hitPoints : int)

Calls the **setHealth** method using the current **healthPoints** attribute minus the passed parameter **hitPoints**.

[HumanResponse.java](#)

HumanResponse(response : String, validAction : boolean)

Calls the setters **setResponse** using the passed **response** parameter and **setValidAction** using the passed **validAction** parameter.

void setValidAction(validAction : boolean) - Assigns **validAction** to the passed **validAction** parameter.

void setResponse (response : String) - Assigns **response** to the passed **response** parameter.

boolean getValidAction() - Returns the **validAction** attribute.

String getResponse() - Returns the **response** attribute.

[Item.java](#)

Same as HW1

[Human.java](#)

Human(name : String, hp : int, emptyBag : Bag)

Calls the setters **setName** using the passed parameter of **name**, **createHealth** using the passed parameter of **hp**, **initBag** with the passed parameter of **emptyBag**, and **initCommandProcessor()**.

void setName(name: String) - Assigns the passed in parameter **name** to the attribute **name**.

void createHealth(hp : int) - Instantiates a new **Health** object using the passed parameter **hp** and assigns the new **Health** object to the attribute **health**.

void initBag(emptyBag : Bag)

Assigns the passed parameter of **emptyBag** to the attribute **bag**; also calls **addItem** with a new **Item** that has the values:
name = "Crowbar", value = 30, weight = 5.

void initCommandProcessor() - Instantiate a new **CommandProcessor** object and assign it to **commandProcessor**.

String getName() - Returns the **name** attribute.

Health getHealth() - Returns the **health** attribute.

Bag getBag() - Returns the **bag** attribute.

boolean isLiving() - Returns the value from calling the **getAlive** method on the **health** attribute.

boolean pickup(item : Item) - Adds the passed **item** to the **bag** using the **addItem** method from the **Bag** class. Returns the result of **addItem**.

boolean drop(item : Item) - Drops the passed **item** from the **bag** using the **dropItem** method from the **Bag** class. Returns the result of **dropItem**.

boolean attack(beast : Beast, item : Item) - Attempts to injure the passed-in **beast** by calling the beast's **injured** method using the passed **item**. Returns the result of **injured**.

HumanResponse processCommand(commands : String, beast : Beast, item : Item)

Processes the command given on the beast, item, or nothing.

- Create **local String** named **response**. Don't assign it to anything (not even null).
- Create **local boolean** named **validAction** and assign it to **true**.
- Split the passed parameter **commands** into a String array named **splitCommands** with a delimiter of a space, (" ").
- Verify that **splitCommands[0]** is a valid command by using the method **validateUserCommand** on the **commandProcessor** variable.
 - If the command is invalid, return a new **HumanResponse** object with **response** as "Invalid game command" and **validAction** as false.
 - If the command is "attack", **splitCommands[2]** must be "with" and the length of **splitCommands** must be greater than or equal to 4. Also, if **splitCommands[1]** is the name of the passed-in **beast** and if the **beast** is living, get the **Item** from the **bag** using the **getItem** method with the String at **splitCommands[3]** as name. Pass that **Item** and the **beast** in to the **attack** method in this class, if the attack was unsuccessful, assign the **response** to "Not a valid weapon" and **validAction** to false.
 - If the command is "pickup", call the **pickup** method in this class on the passed-in **item**. If successful, assign the **response** to "Item added successfully", else **response** is assigned to "Item not added successfully" and **validAction** assigned to false.
 - If the command is "drop" and **splitCommands** is greater than or equal to 2, get the **Item** from the **bag** using the **getItem** method using the String located at **splitCommands[1]**. Pass that **Item** to the **drop** method in this class, if the drop was successful assign the **response** to "Item successfully dropped ", else **response** is assigned to "Item not successfully added" and **validAction** assigned to false.
 - If the command is "help", create an ArrayList of type String and assign it to the results of calling the **getCommands** method on the **commandProcessor** attribute. Using the new ArrayList, concatenate the commands together into the **response** attribute with a new line between each command.
- Return a newly created **HumanResponse** with **response** and **validAction**.

RoomEscape.java

void main(args : String[])

GIVEN

void initGameData()

- Instantiates a new Scanner with System.in to **userInput** attribute.
- Instantiates a new Random with the seed of **1337** to **randomGenerator** attribute.
- Instantiate a new **GameDataReader** Object
 - Call the **getGameItems** method from the **GameDataReader** class and assign the returning ArrayList of type **Item** to the **gameItems** attribute.
 - Call the **getGameBeasts** method from the **GameDataReader** class and assign the returning ArrayList of type **Beast** to the **gameBeasts** attribute.

void initCurrentRoomBeasts ()

Instantiates a new **ArrayList** of type **Beast**, assigning that to the **currentRoomBeasts** attribute and copies all the Beasts stored in the ArrayList **gameBeasts** attribute into the **currentRoomBeasts** ArrayList attribute.

Note: A DEEP COPY NEEDED. A deep copy is when an object has been copied and the copy do not share any memory references that are the same. Thus, using the getters to get attributes from the copied object is a must. You will need to instantiate a new Beast for each Beast in the **gameBeasts** into the **currentRoomBeasts** ArrayList. Link: http://en.wikipedia.org/wiki/Object_copy

Item findItem()

Nearly identical to **scavengeArea**, except you are using an ArrayList for **Items** instead of an array and **randomGenerator** is already created in **initGameData**. This method returns a randomly picked **Item** from the attribute **gameItems**.

Beast findEnemy()

Nearly identical to **scavengeArea** except you are using an ArrayList for Beasts instead of an array and **randomGenerator** is already created in **initGameData**. This method returns a randomly picked **Beast** from the attribute **gameBeasts**.

void displayBagContents()

GIVEN

boolean battle(battlingBeast : Beast)

GIVEN

boolean noBattle(foundItem : Item)

GIVEN

Sample Output (MUST BE THE SAME RANDOM SEEDED)

<p>Enter Character Name: Matthew England Enter number of levels: 1 Entering Room 1</p> <p>Searching area... Searching area... Searching area... Searching area... You have discovered the item Pistol Contents of Bag:</p> <table> <tr> <th>Name</th> <th>HitPoints</th> <th>Weight</th> </tr> <tr> <td>Crowbar</td> <td>30</td> <td>5</td> </tr> </table> <p>Command: pick2 Invalid game command, Try Again: pickup Item added successfully Contents of Bag:</p> <table> <tr> <th>Name</th> <th>HitPoints</th> <th>Weight</th> </tr> <tr> <td>Crowbar</td> <td>30</td> <td>5</td> </tr> <tr> <td>Pistol</td> <td>75</td> <td>10</td> </tr> </table> <p>Command: help pickup drop attack help</p> <p>Contents of Bag:</p> <table> <tr> <th>Name</th> <th>HitPoints</th> <th>Weight</th> </tr> <tr> <td>Crowbar</td> <td>30</td> <td>5</td> </tr> <tr> <td>Pistol</td> <td>75</td> <td>10</td> </tr> </table> <p>Command: attack</p> <p>Contents of Bag:</p> <table> <tr> <th>Name</th> <th>HitPoints</th> <th>Weight</th> </tr> <tr> <td>Crowbar</td> <td>30</td> <td>5</td> </tr> <tr> <td>Pistol</td> <td>75</td> <td>10</td> </tr> </table> <p>Command: Searching area... Searching area... Searching area... You have discovered the item Machete Contents of Bag:</p> <table> <tr> <th>Name</th> <th>HitPoints</th> <th>Weight</th> </tr> <tr> <td>Crowbar</td> <td>30</td> <td>5</td> </tr> <tr> <td>Pistol</td> <td>75</td> <td>10</td> </tr> </table>	Name	HitPoints	Weight	Crowbar	30	5	Name	HitPoints	Weight	Crowbar	30	5	Pistol	75	10	Name	HitPoints	Weight	Crowbar	30	5	Pistol	75	10	Name	HitPoints	Weight	Crowbar	30	5	Pistol	75	10	Name	HitPoints	Weight	Crowbar	30	5	Pistol	75	10	<p>Command: pickup Item added successfully Contents of Bag:</p> <table> <tr> <th>Name</th> <th>HitPoints</th> <th>Weight</th> </tr> <tr> <td>Crowbar</td> <td>30</td> <td>5</td> </tr> <tr> <td>Pistol</td> <td>75</td> <td>10</td> </tr> <tr> <td>Machete</td> <td>40</td> <td>5</td> </tr> </table> <p>Command: drop crowbaa Item not successfully dropped, Try Again: drop crowbar Item successfully dropped Contents of Bag:</p> <table> <tr> <th>Name</th> <th>HitPoints</th> <th>Weight</th> </tr> <tr> <td>Pistol</td> <td>75</td> <td>10</td> </tr> <tr> <td>Machete</td> <td>40</td> <td>5</td> </tr> </table> <p>Command: Searching area... Searching area... Searching area... You have discovered the item Chainsaw Contents of Bag:</p> <table> <tr> <th>Name</th> <th>HitPoints</th> <th>Weight</th> </tr> <tr> <td>Pistol</td> <td>75</td> <td>10</td> </tr> <tr> <td>Machete</td> <td>40</td> <td>5</td> </tr> </table> <p>Command: Searching area... Searching area... Searching area... You have encountered a(n) Vampire Contents of Bag:</p> <table> <tr> <th>Name</th> <th>HitPoints</th> <th>Weight</th> </tr> <tr> <td>Pistol</td> <td>75</td> <td>10</td> </tr> <tr> <td>Machete</td> <td>40</td> <td>5</td> </tr> </table> <p>attack Vampire with pistoll</p> <p>Not a valid weapon Vampire HP: 100 Contents of Bag:</p> <table> <tr> <th>Name</th> <th>HitPoints</th> <th>Weight</th> </tr> <tr> <td>Pistol</td> <td>75</td> <td>10</td> </tr> <tr> <td>Machete</td> <td>40</td> <td>5</td> </tr> </table> <p>attack Vampire with pistol</p>	Name	HitPoints	Weight	Crowbar	30	5	Pistol	75	10	Machete	40	5	Name	HitPoints	Weight	Pistol	75	10	Machete	40	5	Name	HitPoints	Weight	Pistol	75	10	Machete	40	5	Name	HitPoints	Weight	Pistol	75	10	Machete	40	5	Name	HitPoints	Weight	Pistol	75	10	Machete	40	5	<p>Vampire HP: 25 Contents of Bag:</p> <table> <tr> <th>Name</th> <th>HitPoints</th> <th>Weight</th> </tr> <tr> <td>Pistol</td> <td>75</td> <td>10</td> </tr> <tr> <td>Machete</td> <td>40</td> <td>5</td> </tr> </table> <p>attack Vampire with machete</p> <p>Vampire is killed</p> <p>Searching area... Searching area... You have encountered a(n) Shifter Contents of Bag:</p> <table> <tr> <th>Name</th> <th>HitPoints</th> <th>Weight</th> </tr> <tr> <td>Pistol</td> <td>75</td> <td>10</td> </tr> <tr> <td>Machete</td> <td>40</td> <td>5</td> </tr> </table> <p>attack Shifter with hammer</p> <p>Not a valid weapon Shifter HP: 100 Contents of Bag:</p> <table> <tr> <th>Name</th> <th>HitPoints</th> <th>Weight</th> </tr> <tr> <td>Pistol</td> <td>75</td> <td>10</td> </tr> <tr> <td>Machete</td> <td>40</td> <td>5</td> </tr> </table> <p>attack Shifter with pistol</p> <p>Shifter HP: 25 Contents of Bag:</p> <table> <tr> <th>Name</th> <th>HitPoints</th> <th>Weight</th> </tr> <tr> <td>Pistol</td> <td>75</td> <td>10</td> </tr> <tr> <td>Machete</td> <td>40</td> <td>5</td> </tr> </table> <p>attack Shifter with pistol</p> <p>Shifter is killed</p> <p>You have beat RoomEscape!</p>	Name	HitPoints	Weight	Pistol	75	10	Machete	40	5	Name	HitPoints	Weight	Pistol	75	10	Machete	40	5	Name	HitPoints	Weight	Pistol	75	10	Machete	40	5	Name	HitPoints	Weight	Pistol	75	10	Machete	40	5
Name	HitPoints	Weight																																																																																																																														
Crowbar	30	5																																																																																																																														
Name	HitPoints	Weight																																																																																																																														
Crowbar	30	5																																																																																																																														
Pistol	75	10																																																																																																																														
Name	HitPoints	Weight																																																																																																																														
Crowbar	30	5																																																																																																																														
Pistol	75	10																																																																																																																														
Name	HitPoints	Weight																																																																																																																														
Crowbar	30	5																																																																																																																														
Pistol	75	10																																																																																																																														
Name	HitPoints	Weight																																																																																																																														
Crowbar	30	5																																																																																																																														
Pistol	75	10																																																																																																																														
Name	HitPoints	Weight																																																																																																																														
Crowbar	30	5																																																																																																																														
Pistol	75	10																																																																																																																														
Machete	40	5																																																																																																																														
Name	HitPoints	Weight																																																																																																																														
Pistol	75	10																																																																																																																														
Machete	40	5																																																																																																																														
Name	HitPoints	Weight																																																																																																																														
Pistol	75	10																																																																																																																														
Machete	40	5																																																																																																																														
Name	HitPoints	Weight																																																																																																																														
Pistol	75	10																																																																																																																														
Machete	40	5																																																																																																																														
Name	HitPoints	Weight																																																																																																																														
Pistol	75	10																																																																																																																														
Machete	40	5																																																																																																																														
Name	HitPoints	Weight																																																																																																																														
Pistol	75	10																																																																																																																														
Machete	40	5																																																																																																																														
Name	HitPoints	Weight																																																																																																																														
Pistol	75	10																																																																																																																														
Machete	40	5																																																																																																																														
Name	HitPoints	Weight																																																																																																																														
Pistol	75	10																																																																																																																														
Machete	40	5																																																																																																																														
Name	HitPoints	Weight																																																																																																																														
Pistol	75	10																																																																																																																														
Machete	40	5																																																																																																																														

GRADING RUBRIC

50 possible points

If your program does not compile, produce any input/output (I/O) because most of the source code is commented out then your lab will receive a grade of **ZERO**. If your lab has any runtime errors (Such as NullPointerException or ArrayIndexOutOfBoundsException), **you will lose 25 points**.

If you don't have header comments at the top of ALL OF YOUR CLASS FILES, you will receive ZERO points. NO EXCEPTIONS!!!!

Grading Rubric

6 points: Bag.java

5 points: Beast.java

3 points: CommandProcessor.java

5 points: GameDataReader.java

6 points: Health.java

15 points: Human.java

2 points: HumanResponse.java

8 points: RoomEscape.java