

Test #1	“Do you want to control the life of a college student user again? Enter Y for yes and N for no.” (Textgame.java)
	Brief Description (step by step)after the game is over it should ask this question.
	Expected result The console should print “Game over” after the user types “N”, the game will keep going only if the user types “Y”.
	Actual Result The game will keep going regardless of whether the user typed “Y” or “N”
	Status (Passed or Failed) Failed

Test #2	“Use” command (Textgame.java)
	Brief Description (step by step)after the user enters a command, it will ask this question. The user may type “use” after prompted to choose a command.
	Expected result The game should give you an option to use something or give you a message saying you cannot use anything.
	Actual Result The game loops back to the beginning.
	Status (Passed or Failed) Failed.

Test # 3	"Look" command(Textgame.java)
----------	-------------------------------

	Typing this command will tell the user their surroundings.
	Expected result: It should show a text that appropriately responds to this. Maybe showing the environment around the player or objects around him or her.
	Actual Result: Print out "Do you want to control the life of a college student user again? Enter Y for yes and N for no."
	Test Failed

Test #4	"Visit" command (Textgame.java)
	Brief Description (step by step) When a user enters a "visit" command, it will ask this question. The user may type "use" after prompted to choose a command.
	Expected result The console should print options that ask where to visit and options should advance the game.
	Actual Result The game ask the user where they want to visit, however the code breaks if the user picks any option.
	Status (Passed or Failed)failed

Test #5	"check inventory" command (Textgame.java)
	Brief Description (step by step) When user enters a "check inventory" command, it will give the user an array of items to look at
	Expected result The console should print an array of items for the user to see
	Actual Result The console loops "Your choices are as follows: use, look, visit, check inventory, help, Please enter an option: Your choices are as follows: use, look, visit, check inventory, help, Please enter an option:", it never shows any items.
	Status (Passed or Failed)failed

Test #6	“help” command (Textgame.java)
	Brief Description (step by step) When a user enters a “help” command, it will show the user a list of commands.
	Expected result The console should print every command in the game.
	Actual Result The console prints every command in the game
	Status (Passed or Failed)Passed

Test #7	“HouseMain” function
	Brief Description (step by step) ask which command/input the user wants to use. The use command will allow the user to use a plant, door, or dresser. String[] inputs = {"use","look","visit","check inventory","help"};
	Expected result . All commands work and do as the function intended
	Actual Result . The “Check inventory” command never runs if selected.
	Status (Passed or Failed)failed

Test #8	“Houseinit” function
	Brief Description (step by step)Tells the user where they are and ask what they are doing there.
	Expected result . Prints out what is written.
	Actual Result . You walk into your friend's house. What are you even doing in here?:
	Status (Passed or Failed)passed

Test #8	“townSquare” function
	Brief Description (step by step)Tells the user where they are and ask what they want to do. Then it will list commands. Every command should

	work as usual, but the “use:” command should prompt the user to do a specific set of task and each task should print accordingly.
	Expected result . Prints out what is where the user is and all commands should work.
	Actual Result .If the user types “use” they will get an infinite loop of “Which one would you like to go to?: Your choices are as follows: farmer's stand, flower stall, friend's house, Please enter an option: “
	Status (Passed or Failed)failed

#### Cave.cave

Test # 9	“Cave” function
	Brief Description: User goes to the cave with or without items to complete the game. They can look around or take paths to find the treasure. They have the choice to leave to save themselves from failing the mission if they have no items.
	Expected result: “Use” = print out the two options “cave exit or cave path” “Look” = print out what’s around the user, “Visit” = user can go somewhere else, Check Inventory = user can see what they have, Help:user sees valid inputs, cave exit: user leaves cave, cave path: print out the line from 51-54,
	Actual Result: Typing cave exit or cave path is treated as an invalid input. It prints out “Your choices are as follows...” twice. Prevents the user from going forward  Test Failed

Test # 10	“Home” function
	Brief Description: User goes to home. The user can look around the house for items, or use the items that are available at that situation to help the user proceed the game. I did test this Home class, but since it is a Home constructor, it won't be able to print out all the code inside this class. So I try to change the method by using  “ public static void Home (String NAME, String place, String command, String[] inputs) { ... }.

	<p>Then I tested it by creating another class that has the same package as every class of this game. I typed “</p> <pre>public static void main (String[] args) {     Home.Home (“David”, “Home”, “use”, args); }</pre> <p>Then it print out the statement like this:</p> <p>“David is at their home... It’s a dump. What does David wanna do?! Your choices are as follows:</p> <p>Please enter an option: (I type every single of commands as well) Your choices are as follows:</p> <p>Please enter an option: “</p> <p>It always goes back over again for every command that the user types.</p>
	<p>Expected result: The user should be able to use all the commands when they are home such as visit, help, check inventory, look, and use. Be able to look around the house, and use all the items that are available depending on what kind of items the user has in their inventory. So if they use the right item, they can proceed to the next step of this game.</p>
	<p>Actual Result: It still prints out the first statement and asks the user to choose an option that the user wants to. But it will go back and ask the user again to choose an option. So it won't let the user go forward to the next step of the game. Prevent users from using an item available at Home.</p> <p>Test Failed</p>

Test # 11	“Check inventory” function(Inventory.java)
	<p>Brief Description: Should show what the user has in their inventory when the program runs The user should only have to input their name.</p>
	Expected result: “Name” has nothing.
	<p>Actual Result: “Name” has nothing.</p> <p>Test Passed</p>

Test # 12	“addItemToInv” function(Inventory.java)
	Brief Description: Should add whatever item is inputted into the text If the user inputs “item”.
	Expected result: “Name” has “item”.
	Actual Result: “Name” has “item”.  Test Passed

Test # 13	“RemoveInventory” function(Inventory.java)
	Brief Description: Should remove whatever item is inputted into the text If the user inputs “item”.
	Expected result: “Name” has nothing.
	Actual Result: “Name” has nothing.  Test Passed

Test # 14	“ClearInventory” function(Inventory.java)
	Brief Description: Should remove all items is inputted into the text If the user inputs “item” and then inputs “moreitems”.
	Expected result: “Name” has nothing.
	Actual Result: “Name” has nothing.  Test Passed

Test # 15	“libraryFront” function “look” command(Library.java)
	Brief Description: Upon accessing the function utilizing: Library.libraryFront(“Henry”); And entering the “look” command an infinite loop occurred.
	Expected result: A description of player surroundings and able to interact with the environment through other commands.
	Actual Result: Infinite repeating of “look” command print statements.

	Test Failed
--	-------------

Test # 16	“libraryFront” and “libraryBack” function calling checkInventory(Library.java)
	Brief Description: Upon calling libraryFront utilizing: Library.libraryFront(“Henry”); / Library.libraryBack(“Henry”); And typing in the command “check inventory” nothing was printed out.
	Expected result: printing out the player’s inventory
	Actual Result: Nothing printed out.  Test Failed

Test # 17	“libraryFront” and “libraryBack” function “help” command(Library.java)
	Brief Description: Upon accessing the function utilizing: Library.libraryFront(“Henry”); / Library.libraryBack(“Henry”); And entering the “help” command an infinite loop occurred.
	Expected result: A list of commands the player could utilize and then ability to enter a different command..
	Actual Result: Infinite repeating of “help” command print statements.  Test Failed

Test # 18	“libraryFront” and “libraryBack” function “use” command(Library.java)
	Brief Description: Upon accessing the function utilizing: Library.libraryFront(“Henry”); / Library.libraryBack(“Henry”); And entering the “use” command an infinite loop occurred.
	Expected result: A description of player surroundings and interaction with it then ability to enter a different command. Add map to inventory.
	Actual Result: Infinite repeating of “use” command print statements. Unable to utilize the map added to inventory or verify it’s addition to the

	<p>player's inventory.</p> <p>Test Failed</p>
--	---

Test # 19	"libraryEntrance" function(Library.java)
	<p>Brief Description: Upon accessing the function utilizing: Library.libraryEntrance("Henry"); Print statements are activated.</p>
	<p>Expected result: A description of the library entrance and option to go deeper into the library</p>
	<p>Actual Result: A description of the library entrance and a bizarre player input required to venture farther into the library.</p> <p>Test Passed?</p>

Test # 20	"libraryBack" function "use" command(Library.java)
	<p>Brief Description: Upon accessing the function utilizing: Library.libraryBack("Henry"); And entering the "use" command followed by entering the name of the object "chair" and "books"</p>
	<p>Expected result: Interactions with the objects and the ability to continue the game from there.</p>
	<p>Actual Result: No output by the program and inability to continue the game.</p> <p>Test Failed</p>

Test # 21	"libraryBack" and "libraryFront"function "visit" command(Library.java)
	<p>Brief Description: Upon accessing the function utilizing: Library.libraryFront("Henry"); / Library.libraryBack("Henry"); And entering the "visit" command.</p>
	<p>Expected result: Ability to visit other locations</p>
	<p>Actual Result:Ability to visit other locations.</p>



	Test Passed
--	-------------

Test # 22	"libraryBack" function "look" command(Library.java)
	Brief Description: Upon accessing the function utilizing: Library.libraryBack("Henry"); And entering the "look" command an infinite loop occurred.
	Expected result: A description of player surroundings and able to interact with the environment through other commands.
	Actual Result: Infinite repeating of "look" command print statements.  Test Failed