

Introduction to Programming

CMPT 120 • Fall 2014

-Final Project - game v1.0 - 200 points

Goals	To finish development of your semester-long project: a text adventure game in the spirit of The Hitchhikers Guide to the Galaxy, Planetfall, and Zork. Also, to show off your expertise in using Software Development Best Practices , as well as Git.
Instructions	<p>Fix anything that was incorrect or incomplete with your prior projects. (Commit. Push.) Then, beginning with a perfect implementation of the prior version of your game, implement the following new features:</p> <ul style="list-style-type: none"><input type="checkbox"/> Add a puzzle element to the gameplay so that it's necessary for the player to solve the puzzle to "win" or at least complete the game. [75 points]<ul style="list-style-type: none">• Perhaps the player needs to acquire certain items to progress to the final location.• Maybe some locations need to be encountered in a specific order.• It could be something that can only happen after a given number of moves, or after a certain score is achieved by one or both of the methods above.• Conceivably it's some combination of all of these. Be creative. But be sure that the puzzle makes sense in the context of the game and that it's solvable by the player.<input type="checkbox"/> Redo your navigation one final time, this time with a matrix; no more if-else, no more switch-case. Write a function that takes the current location and direction moved as parameters and returns the new location (or the same location if that was an invalid move). Use that function to run all player navigation through the game. [75 points]<input type="checkbox"/> All non-navigation requirements from all projects 2, 3, 4, and 5 perfectly implemented [50 points]<input type="checkbox"/> Software Development Best Practices ubiquitous in your source code [- ∞ if not]
Advice	<p>Test, test, and test again. Then test some more. When you think you've tested enough, go back and test again. Then get someone else to test for you while you test theirs. Rinse and repeat.</p> <p>Push your work to your Git repository early and often. While you're in there ...</p> <ul style="list-style-type: none">• Be sure to write meaningful commit messages.• Practice using <i>diff</i> to see the differences between successive versions of your code.• Practice reverting to an earlier version so that you'll have that option in the future. <p>Don't forget to test. A lot. Really. (Rilly.)</p>
Submitting	<ol style="list-style-type: none">1. Push your work to your Git repository before the class in which it is due.2. Print and staple your source code before class and hand it in at the start of the class in which it is due. Remember to include your name, the date, and the assignment in the (copious, meaningful, and accurate) comments in your code.