

Software Development One

CMPT 220 • Spring 2014

- Project Four - 125 points

Goal	Still more fun with your interactive fiction adventure games.
Instructions	<p>Design, develop, and thoroughly test the final release version of your very own, original, interactive fiction game in the spirit of Adventure and Zork.</p> <p>Everything from projects two and three are required. If there were any deficiencies in your prior projects you must fix them for this project. Repeated deficiencies will result in repeated point loss. Repeatedly.</p>
Deliverables	<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 complete the game.<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"/> Keep track of every locale the player visits. When they win or quit the game, after displaying their number of turns, money, score, and skill ratio, ask the player if she would like to review her walk-through <i>forward</i> or <i>backward</i>.<ul style="list-style-type: none">▶ If the player chooses to review her adventure <i>forward</i>, list each location she visited in the order she visited them. Use a queue data structure to keep track of this path during the game and to replay it at the end.▶ If the player chooses to review her adventure <i>backward</i>, list each location she visited in reverse order, as if to "walk her backwards out of the game". Use a stack data structure to keep track of this path during the game and to replay it at the end.<input type="checkbox"/> About the <i>Magick Shoppe</i>:<ul style="list-style-type: none">▶ Once, at initialization . . .<ul style="list-style-type: none">- After reading the list of items from <code>magicitems.txt</code>, a copy of which can be download from our web site and included in your code directory, declare a fixed-length array and copy the items into that array from the list.- Sort the array.▶ Every time the player enters the shop . . .<ul style="list-style-type: none">- Prompt the player to enter the item they want to buy.- Binary Search search the list of items.- If found, display the item that matches what the player typed, else display a message that you don't have it and let them search again or leave.- If the player has enough money, let her buy the item and add it to her inventory.
Source Code	<input type="checkbox"/> Your code must separate structure from presentation, be professionally formatted, use and demonstrate best practices, and make me proud to be your teacher.
Submitting	Push to your GitHub repository early and often. Write great commit messages. Push the final version before the class in which it is due and e-mail me a link to your repository. Remember: code quality, comments, and neatness count.