

# Software Development One

CMPT 220 • Spring 2014

## - Project Three - 125 points

---

Goal	More fun with your interactive fiction adventure games.
Instructions	Design, develop, and thoroughly test the next release version of your very own, original, interactive fiction game in the spirit of Adventure and Zork..
Deliverables	<ul style="list-style-type: none"><li><input type="checkbox"/> Everything from project two is required. If there were any deficiencies in your project two you must fix them for this project. Repeated deficiencies will result in repeated point loss. Repeatedly.</li><li><input type="checkbox"/> Use a linking objects for location navigation; no matrix, no if-else, no switch-case.</li><li><input type="checkbox"/> Your game must still have at least eight (8) different locations, one of which is a <i>Magick Shoppe</i>.</li><li><input type="checkbox"/> Modify your Location base class (its subclasses should not need to be modified) to support navigation through linking objects.</li><li><input type="checkbox"/> Sprinkle some form of money throughout the locations so that the player can buy stuff at the <i>Magick Shoppe</i>.</li><li><input type="checkbox"/> About the <i>Magick Shoppe</i>:<ul style="list-style-type: none"><li>▶ Once, at initialization . . .<ul style="list-style-type: none"><li>- Read the list of items from magicitems.txt, a copy of which can be download from our web site and included in your code directory.</li><li>- Make the prices random.</li><li>- Store the items and their prices (as objects) in a linked list.</li></ul></li><li>▶ Every time the player enters the shop . . .<ul style="list-style-type: none"><li>- Prompt the player to enter the item they want to buy.</li><li>- Linear search the list of items.</li><li>- 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.</li><li>- If the player has enough money, let her buy the item and add it to her inventory.</li></ul></li></ul></li></ul>
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 <b>before</b> the class in which it is due and e-mail me a link to your repository. Also, print your source code, staple it all together, and hand it in during the class in which it is due. Remember to include your name. Remember also that neatness counts.