

# Software Development One

CMPT 220 • Spring 2014

## - Project Two - 125 points

---

Goal	I want you to have fun with developing an interactive fiction adventure, this time in Java!
Instructions	Design, develop, and thoroughly test the 1.0 release version of your very own, original, interactive fiction game in the spirit of Adventure and Zork, in <b>Java</b> .
Deliverables	<ul style="list-style-type: none"><li><input type="checkbox"/> Display the title and the first location of your game when it starts.</li><li><input type="checkbox"/> After every move, display ...<ul style="list-style-type: none"><li>• the current location</li><li>• the score</li><li>• the number of moves made so far</li><li>• achievement ratio: points / number of moves</li><li>• directions in which the player can move from the current location.</li></ul></li><li><input type="checkbox"/> Use a matrix (two-dimensional array) for location navigation; no if-else, no switch-case; no linking objects (yet).</li><li><input type="checkbox"/> Write a function that takes <i>current location</i> and <i>direction moved</i> as parameters and returns the new location (or the same location if it's an invalid move).</li><li><input type="checkbox"/> Take text input in a case-insensitive manner.<ul style="list-style-type: none"><li>• Valid commands are the directionals <b>n, s, e, w</b>; <b>i</b> for inventory; <b>h</b> for help, <b>t</b> for take, <b>m</b> for map, and <b>q</b> for quit. (Maybe <b>d</b> for dance, if you're into that. Feel free to add more. Make it fun.)</li><li>• If the player enters <b>m</b>ap but has not picked up the map, do not display it. (See below.)</li><li>• If the player enters an invalid command then say so and explain the valid commands.</li></ul></li><li><input type="checkbox"/> Your game must have at least eight (8) different locations, one of which is a <i>Magick Shoppe</i>.</li><li><input type="checkbox"/> Develop a Location class {id, name, description, item[], toString()}</li><li><input type="checkbox"/> Develop a subclass of Location to represent some specialization of Location. (Space, Water, etc.)</li><li><input type="checkbox"/> Instantiate a few instances of Location() and a few of its subclass for each location in your game.</li><li><input type="checkbox"/> Store the game location instances in an array.</li><li><input type="checkbox"/> Write a routine to display each location using the current location and the array of location instances. (Remember, your subclass IS-A Location.)</li><li><input type="checkbox"/> When the player enters the <i>Magick Shoppe</i>, display a list of magic items available for sale.</li><li><input type="checkbox"/> Write a method to display a map of your game environment. ASCII art is fine.<ul style="list-style-type: none"><li>• Consider the map to be an item and place it in the game's first location.</li><li>• Once the player takes the map, enable a <b>m</b>ap command to display the map.</li><li>• To be really cool (and for extra credit), denote the current location graphically in the map.</li></ul></li><li><input type="checkbox"/> Keep score. Add five (5) points <b>the first time</b> each time the player goes to a location.</li><li><input type="checkbox"/> Develop an Item class {id, name, description, toString()}</li><li><input type="checkbox"/> Instantiate an instance of Item() for each item that can be found in your game (at least four (4) of them, please) and then store zero or more at each location.</li><li><input type="checkbox"/> Create an array to hold the player's inventory.</li><li><input type="checkbox"/> Allow the player to take items and put them in their inventory. Be sure that your location descriptions are accurate regarding whether or not an item is present to take.</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. Neatness counts.