**Development Report**

Initial Research

When planning, and researching my algorithm I ended up playing a few text adventure games before attempting the assessment. I played these games mostly for examples of different input parsers and commands. The games I played gave players the freedom to type commands, and could handle complex commands from intelligent parsers.

Games used for research

* Zork
* Jigsaw
* Galatea

When playing the games above all command parsing was very similar, the main format used was a verb then a noun for example “Take Rock” or “Attack Enemy”. I noticed from looking at these games that commands for movement and inventory had more than one input which did the same thing, for example inventory you could type “i” or “inventory”. I wanted to add this to my algorithm as it didn’t force player to write long commands constantly, instead of players typing “Inventory” every time they wanted to check what item’s they have they could just type “I”.

Input Parsing

When deciding on my input parsing I first tried using the “Contains” method which would check to see if the users input contained part of a command (Dotnetperls.com), (msdn.microsoft.com). At first this method worked and there were no problems, however when I started adding more items and commands to the game I had a big problem. If I typed a word that contained “use” or “take” It would trigger those commands and not the command I wanted. To fix this problem I decided to change the input parsing into a string array, this allowed me to check each word separately.

When I finished my input parsing I then changed my focus to player inventory. To implement this I created variables for the item names and descriptions and created a list for the inventory. The way

The next idea I tried for my game, was the ability to add exits once a certain item was used in game. For example, in my game you need to brake a lock with a crowbar, if you don’t use the crowbar on the lock you that exit will not be available.