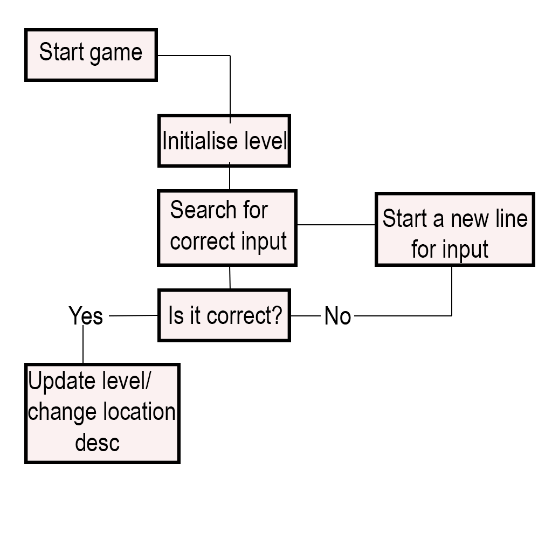
**Report**

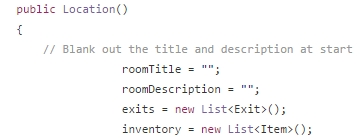
In this report I will explain the iterations I used throughout creating a text adventure game. I will also detail my algorithm development throughout the process.

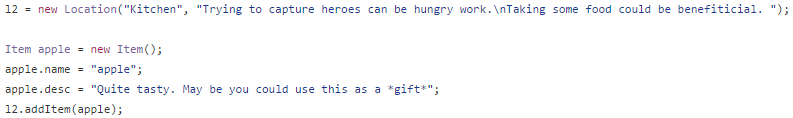
The game I used for reference material for the production of my game is “Zork 1.” From playing Zork 1 I found that it involved a lot of trial and error within while telling a story. However, for the production of my game I iterated that and allowed the player to see what was in the location and what routes were available to prevent a large amount of trial and error.

**Figure 1**: The algorithm I used within my text adventure.

**Location class/ game class**

Once the player starts the game, it will load the location description, location and items available in that area. For my location class data, I used strings to display the room title and descriptions. (As shown in **Figure 2**). The strings held the data within the game class with the room titles and descriptions. The locations were



**Figure 2**: The first set of quotation marks indicates the room title, as shown in the second image that will be displayed as “Kitchen”. The second set is for the description.

For the input parsing I used a string array system for the inputs. I found this easier in development for my text adventure as my game did not require many inputs. If, however I created a much larger build then I would look at a different option, possibly a contains method (List) which would check the player’s inputs and try to match it with a command I created.

A number of guides recommended to use it but I stuck with the string array system as I was comfortable using it and worked just as efficiently.

My player inputs focused on verbs like “take”, “use”, “combine” and “examine”. In **Figure 3a** it shows that the first input “take” is a command. Using that command allows the player to take the items in that area. By typing take and then the item name, i.e. leaf blower it would then be added to the player’s inventory and removed from that area. “Use” works the same way but only works if the item exists in the player’s inventory, it will then remove itself from the inventory and update the locations description. “Combine” allowed 3 inputs. The first to be the combine command and then the other 2 inputs for two item names. In my text adventure I implemented this with an apple and poison. Once combined the item would then be used later on to finish the game, to also give the player knowledge in how to perform this in Location 4 (**Figure 3b**) there is a piece of paper which will give instructions for the player to make it. Location 4 also implements the “examine” input to allow the player to look at the description of the items. I implemented descriptions for each item with hints so players can then later use them in the correct location.

 **Figure 3a**: The input length allows two inputs by the player. 0 being the word “take” and the word followed would be the item in the area.

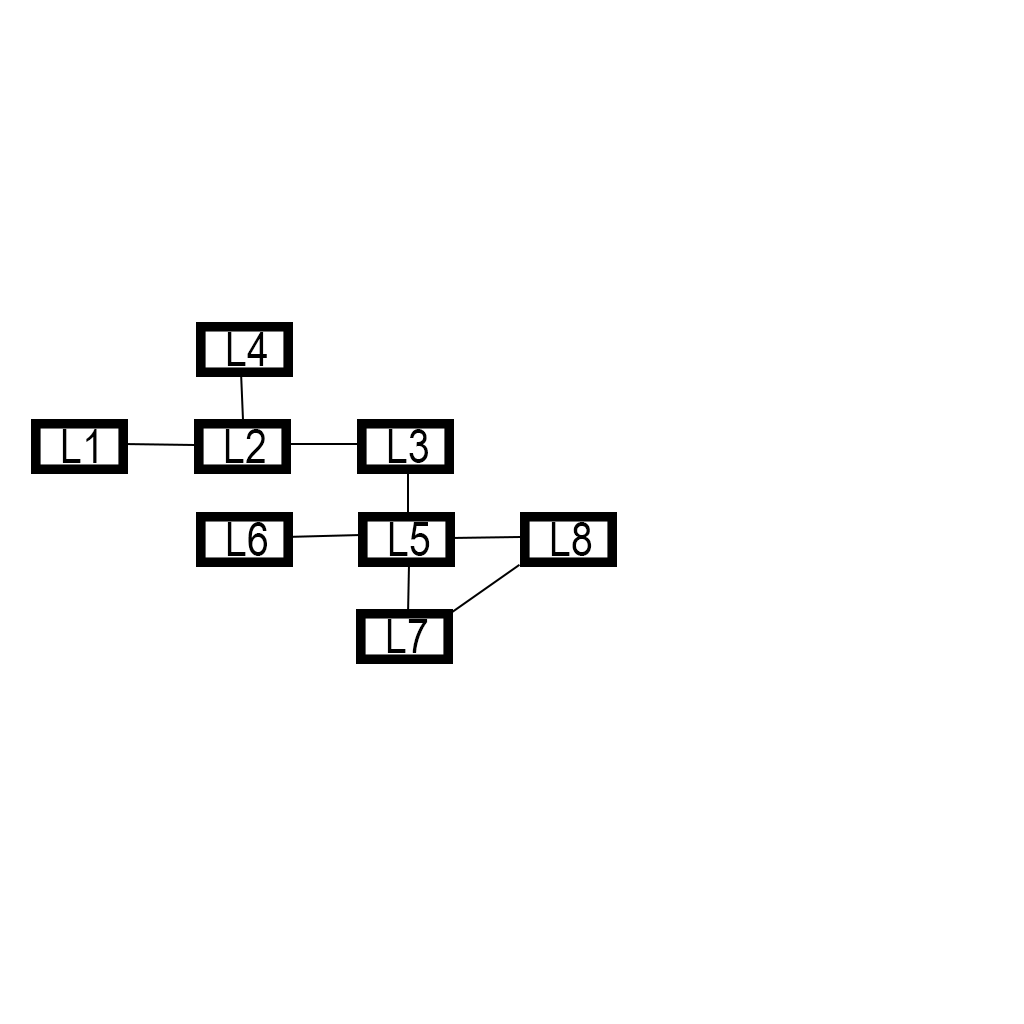


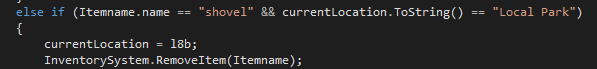
**Figure 3b:** Location 4 (Cupboard) which once the player inputs “examine” it will give information to the player.

**Exit Class**

For the exit class a public enum was used within the class for the directions and a public static string to allow the player to use shortcuts for directions, i.e. “n” for North. Despite the term “undefined” not being used in the directions, it was recommended to keep the constant there for the enum to start at the value of 0 rather than 1. (Enum, msdn.microsoft.com).

The map design for my game consisted of rooms having exits to other locations shown in **Figure 4a**. Most locations only had one exit however, some locations like Location 8 unlocks a new exit once items have been used in sequence. Normally remaining in location 8 will only have one exit which is left, but once the items have been used the location will update and a new exit is available (**Figure 4b**). This will unlock a route to a variant of a location (Location 7b). How I implemented this is as of follows,

**Figure 4a**: The text adventure map. Some exits only become available after items have been used.



**Figure 4b**: Once the shovel is used in the Location titled “Local Park” it will remove the shovel and update the current location to “Dug up Local Park”.

**The Inventory System**

Originally, I used an array system for my old inventory set up although I came across a lot of troubles as I was continuously adding items. So I Iterated and changed it to a list and worked efficiently. Once a player has input “Take”, “Use” or “Combine” followed by item(s) names then it will update the inventory system. The player can then check their inventory for items by inputting “I”, “inv” or “inventory” so that they can view the changes.

**Iterations**

Throughout this project I came up with multiple ideas on what to implement within my text adventure. I previously wanted to add a combat system but when trying to create it I came across multiple issues. This would have also caused me to implement far more classes including enemy health, enemy variants, enemy damage and the same with the player too. For the story I was aiming for it didn’t suit it so I stuck with an adventure where the player interacts with the world more so than focusing on combat.

**Conclusion**

Overall I thought this project went well as I started learning a lot of c# through tutorials and lecturer help and learnt a lot of different ways you can programme a certain class, i.e. Location directions can use enums or a public const string array system as it’s a fixed amount.

**References;**

**Documentation;**

Contains List method: <https://msdn.microsoft.com/en-us/library/bhkz42b3(v=vs.110).aspx>

Arrays vs Lists: <http://stackoverflow.com/questions/434761/array-versus-listt-when-to-use-which>

Enums: <https://msdn.microsoft.com/en-us/library/sbbt4032.aspx>

Enums Example: <http://stackoverflow.com/questions/3519429/what-is-main-use-of-enumeration>

Public const string arrays: <http://stackoverflow.com/questions/5142349/declare-a-const-array>

String array input: <http://stackoverflow.com/questions/230454/how-to-fill-an-array-from-user-input-c>

Variable guide: <https://www.tutorialspoint.com/csharp/csharp_variables.htm>

Removal from a list: <https://msdn.microsoft.com/en-us/library/cd666k3e(v=vs.110).aspx>

**C# Video Tutorials;**

Visual Studio Learning the basics: <https://www.youtube.com/watch?v=Yj0G5UdBJZw>

Visual Studio for Beginners: <https://www.youtube.com/watch?v=lisiwUZJXqQ>

Creating a Text Based Adventure: <https://www.youtube.com/watch?v=vAo2aTfp3cg>

C# Tutorial: <https://www.youtube.com/watch?v=gfkTfcpWqAY>

**Games;**

Zork: <https://textadventures.co.uk/games/view/5zyoqrsugeopel3ffhz_vq/zork>