

Report: Task 9 – Game Data Structure

Author: Khang Trinh - 102118468

Task goal: to figure out what data structure would fit for making an inventory system for Zorkish adventure.

The current design of the inventory system in Zorkish Adventure:

- Currently the game spec doesn't have a list of items the game will be having when playing, so the size of the container will need to be adaptable
- As the item count increases, because the game is text based, there will be a need for sorting alphabetically to make things easier to find
- There's currently no need for items to be inserted in a specific "slot" of the inventory
- The game will need to know what item the player wants to get, and which slot in the inventory it needs to search to get that item

Expectations: Will be using an array for this. It's as easy as going to the indexed slot and get the item out

There are a few containers that can be used to create the inventory system

Based on the diagram provided above, if we were to use:

Array:

- Advantage: Access to any element is $O(1)$. To find the item, you need to know its index
- Disadvantage: Has fixed size, changing size at runtime requires creating a new array with new size then copy old contents over (operation is $O(n)$). Things like inventory space upgrades will likely cost a lot of performance as item count increases

List:

- Advantage: Dynamic growth
- Disadvantage: Access to any element except the first and last requires traversing through the entire container, operation is $O(n)$

Vector:

- Advantage: Is a combination of array and list, able to access via index and can resize at runtime if need to, can contain any type of object (suggests can be sorted by item type), and can also sort those objects using custom sorting policy

Map:

- Advantage: Store element using key - value combination (ie. "Item name" – item count). Keys are also stored sorted, so the items will be sorted automatically on insert
- Disadvantage: Key has to be unique. Therefore, you won't be able to sort items by item type, or if you do try to, it would seem to be more complicated than using a vector (haven't tried yet, this is only speculation)

We will be using a map for this. Looking at this first glance, being able to find things using their name is better than having to find them by remembering their index value. Even though sorting by item type doesn't seem to be viable using a map, as compared to using a vector, if we were to have 50 items displayed in the inventory, it would be better to see them listed alphabetically anyway.

Main links that helped a lot

1. Everything one needs to know about all the available containers in C++
<https://en.cppreference.com/w/cpp/container>
2. A general rule of thumb for choosing the right container
<https://embeddedartistry.com/blog/2017/08/23/choosing-the-right-stl-container-general-rules-of-thumb/>
3. A flow diagram to further help with choosing the right container
<https://stackoverflow.com/questions/471432/in-which-scenario-do-i-use-a-particular-stl-container>
4. More about maps - the container we'll be using in this report
<https://youtu.be/nPSDR5nZzHA>

