COS30031 – Task 9 – Short Report

102564760 Ryan Chessum

C++ Contains many different data structures that we can use to store and retrieve data with. In this report I will look at the advantages and disadvantages of lists, maps, deques and vectors to determine which one would be best to implement an inventory system in Zorkish.

List

Lists are doubly linked lists that use an iterator to access elements. Unlike arrays memory for elements in a list are not in order in memory meaning they can store a large amount of data.

Advantages:

- More efficient for storing larger objects or more elements as data stored is not contiguous in memory.
- Elements can be inserted or deleted at any point in the list

Disadvantages:

- Must use an iterator to access elements in the list rather than a regular index
- You must cycle through the list every time you want to access an element at a position

Deque

A deque is a data structure that has the features of both a stack and a queue. Data can be inserted or obtained at each end.

Advantages:

- Data at each end of the deque can easily be inserted and obtained.
- Can be used like a stack if we need
- Can be used like a queue if we need
- The storage size of the deque can be expanded and contracted as needed
- Elements can be accessed with an index

Disadvantages:

- Slightly slower performance wise compared to a vector. Indexed access requires performing two pointer dereferences compared to the vectors one.

Map

Maps can be used to store a key and value pair. Like lists they are great for storing larger objects as memory locations do not have to be sequential.

Advantages:

- Data could be obtained with a string key for say an item name instead of an index

- Can store pairs of data
- Can store larger objects

Disadvantages:

- Cycling through each set of data requires more work than other data structures that use an index.

Vector

A vector is functionally similar to a classic array. It stores items contiguously in memory and elements can be accessed by an index. However

Advantages:

- The storage size of the vector can be expanded and contracted as needed
- Simple and easy to use
- Elements can be accessed with an index

Disadvantages:

- Data stored in contiguously in memory.
- Cannot store data of too large a size

Conclusion

To implement an inventory system, I think a Vector would be the best fit. Our inventory only has to be very basic; a vector provides easy access and storage of data that can be expanded and contracted as we need. The data we need to store isn't big enough to justify the need for a list or a map. We also don't need to access our data as if it were in a stack or a queue and can just add new items to the end of our vector.