**Spike:** Spike No.9

**Title:** Task 9 – Game Data Structure

**Author:** Khang Trinh - 102118468

**Goals / Deliverables:**

The goal of this spike is to help the developer learn how to figure out the data type they should be using in a given scenario (in this case, an inventory system) and how to find info about using them.

**Technologies, Tools, and Resources used:**

* Visual Studio 2017

**Useful Links:**

1. Everything one needs to know about all the available containers in C++

<https://en.cppreference.com/w/cpp/container>

1. 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/>

1. 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>

1. More about maps - the container we’ll be using in this report

<https://youtu.be/nPSDR5nZzHA>

**Tasks undertaken:**

## Step 1: Figure out what you will be doing with your data in the container

Ask yourself how your data will be used. Here are a few example questions to consider:

* How big is your container going to be? Is it fixed? Infinite? Currently unknown?
* Does your data require to be inserted in a specific slot (front/back/etc.) in the container?
* How are you going to access the data? Random? Key? Index?
* How fast/slow do you need the search for the data to be?

IMPORTANT: It’s crucial that you also think about how expandable your solution can be. What works for you today might not work for you later down the line when you start to have more needs regarding how your data can be used. Can any of the conditions considered above potentially change in the future?

## Step 2: Find out which container works best with what you need

One way to find out which container you need is to know all of the possible containers that could fit your needs. With this, primary source is the most reliable (refer to link 1). When you come across a new container, make note of its properties (size restriction, how it handles data insertion/removal/accessing, unique features) and consider how it could meet your needs.

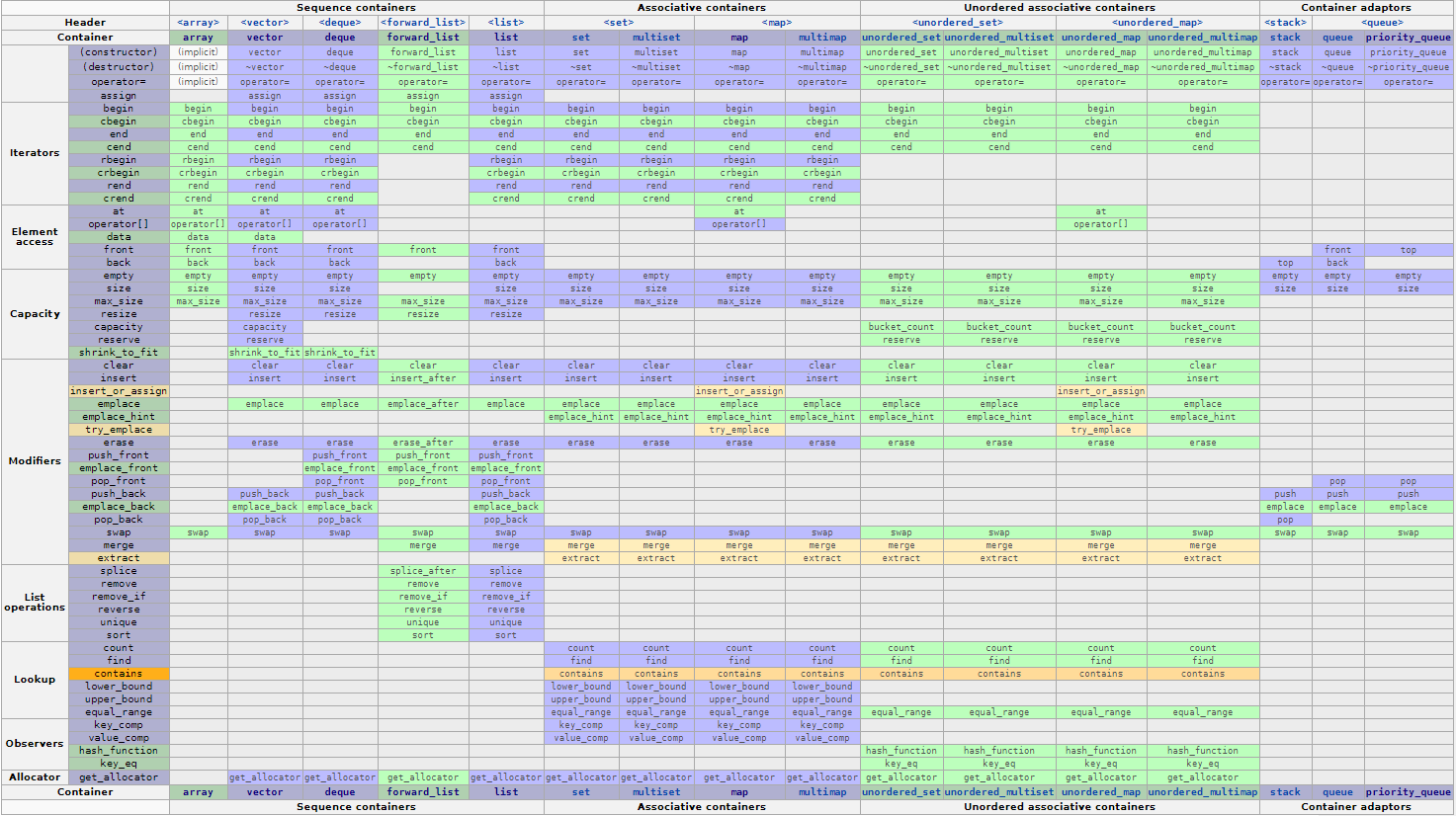


Fig 1. Table showing all the features available for each type of container on cppreference

Another way to find out which container you need is to ask what others use/would use if they were in your situation. You might get mixed answers, but at least that could give you some pointers in where you could be looking. For example, if people you ask tell you that using a vector or a map would be the best way to create an inventory system, then those two should be your top priority to look into.

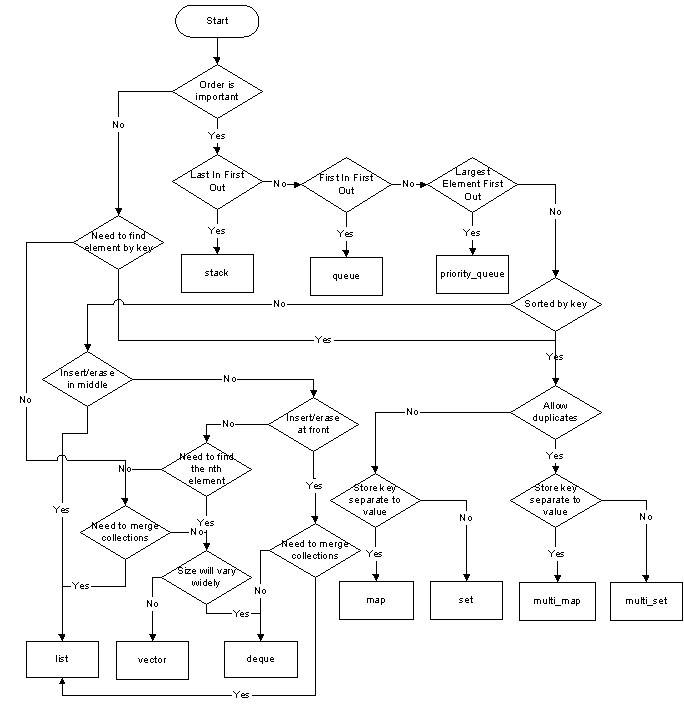


Fig 2. A flow chart that could help you decide which container to use

## Step 3: Learn and implement!

Say you’ve chosen to use a map for your inventory, but you have no idea how to use it. All containers have a way to declare, add/remove/modify things. Google your way through while keeping this in mind until you’ve learned how to implement the container of your choice and make use of its strong points.

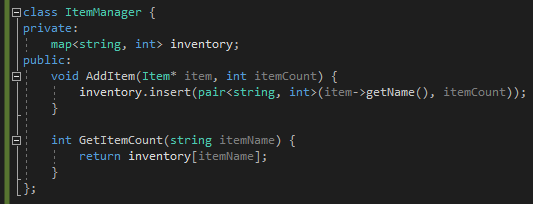


Fig . An example of an inventory system