

**Spike:** Task 9**Title:** Game Data Structures**Author:** Luke Valentino, 103024456**Goals / deliverables:**

The goal was to design and create an inventory system that is capable of storing, viewing, adding, and removing items.

For example: UML diagram, code, reports

- Code see <https://github.com/LukeValentino138/COS30031-2023-103024456>
- Spike Report
- Short Report.

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

List of information needed by someone trying to reproduce this work

- Swinburne Games Programming Lectures
- SDL version 1.2.3.4
- C++ Containers: <https://en.cppreference.com/w/cpp/container>
- C++ Map: <https://en.cppreference.com/w/cpp/container/map>
- W3School C++

**Tasks undertaken:**

List key tasks likely to help another developer

This section should resemble a tutorial – the goal is to allow another coder to reproduce your work following these steps.

Eg: (Good)

- Download and install Visual Studio
- Download and install DirectX
- Configure VS Project File to point to the DX lib folder
- Compile sample code

Not: (Bad)

- Read the source code
- I had some trouble with SDL, so I spent a couple of weeks doing other spikes
- Run code
- Write Spike Report

**Tasks Undertaken:**

- Create empty inventory class.
- Declare a map with a structure of `<string, int>` as private.

- Create addItem (to add items to the inventory). This function takes an item name and quantity and adds it to the map.
- Create printInventory (this is used for viewing the inventory). Iterate through the map, retrieving and printing each item.
- Create removeItem (to remove items from the inventory). Search the map for a specific item, if it is found, minus the quantity. If it is not found nothing should happen.

**What we found out:**

Describe the outcomes, and how they relate to the spike topic + graphs/screenshots/outputs as needed and should move on.

Using a map data structure, I was able to successfully create an inventory system with adding, removal, and viewing capabilities. The map was deemed the most appropriate data structure as outlined in the short report.