

Task 10 - Spike: Game Data Structures

CORE SPIKE

Context: Game developers will often encounter a variety of different types of data and access/usage scenarios, even in a single project. It is essential, therefore, that developers be aware of the different data types available to them, their advantages and disadvantages, and suitability for different purposes.

Knowledge/Skill Gap: The developer is not familiar with common data types, their various strengths and weaknesses, and their applicability in common game scenarios .

Goals/Deliverables:

[CODE] + [SPIKE REPORT] + [SHORT REPORT]

1. Research and evaluate four different data structures that could be used to create the player inventory for the Zorkish game. At a minimum, you must show your awareness of advantages and disadvantages for this application. Document your evaluation criteria and results in a short report.
2. Using your decision (as documented in your short report), create a working inventory system demonstration program. Your work must demonstrate (bug free) inventory access, addition and removal.

Note: The short report is separate from the spike report. Ask your tutor about if you aren't sure what to put in each. Your work won't be accepted unless you have all deliverables .

Recommendations:

- A nice overview of different data structures is available in the C++ STL is here: <http://en.cppreference.com/w/cpp/container>. You may use other libraries, but the STL is recommended.
- Keep the short report SHORT - very focused and concise. No padding or fluff!
- If you do not yet have a specific Player Character class/object for Zorkish, now might be a great time to create one.
- In this case the demonstration program does not technically need to be part of your other Zorkish code features that you've created so far.
- Include a map