Task 12 – Spike: Game Graphs from Data

CORE SPIKE

Context: Selecting appropriate data structures and representations for game information are both critical performance and development issues for game programmers. Graphs are a general data structure with many applications. Developers should be able to take advantage of graphs data structures in their game implementations.

Knowledge/Skill Gap: The developer is not familiar with the use of graphs data structures for representation of game world composed of locations and connections.

Goals/Deliverables:
[FORMAT DESIGN] + [CODE] + [SPIKE REPORT]

Extend the Zorkish program you created in Tasks 9 and 10 (spikes) to include the loading of an Adventure text file and the representation of location and connection information as a graph data structure. Refer to the game specification document on the unit website for details. Your implementation at this stage does not (yet) need a complex command processor. It only needs to support “go” commands in order to show the graph structure (move around the graph world locations).

Create a Zorkish game that demonstrates the following:

1. Specify on paper (REQUIRED) a text-file format that represents a Zorkish game “Adventure” details. Specifically, it will need to include (at this stage) world locations, location details (name, description etc.) and connections to other locations. You can include other details in your design, but we only need locations and connections for this spike.
2. Be able to load an “Adventure” file from the “Select Adventure” game state.
3. Store the locations and connections as a graph data strucutre in your program.
4. Implement ONLY a basic “go” command with directions that map onto your graph.

NOTE: Do NOT implement a full command processor/pattern! Not needed yet.
NOTE: You must implement more than North, East, etc. This is NOT a grid world – any direction is possible!

Recommendations:

- Don’t worry about implementing objects or other entities in the world for this spike. Just focus on locations and connections and some basic move commands
- Don’t worry about a command processing system (that's another spike). This spike is about graph representation of game locations.
- Read the game specifications again. Clearly identify the minimum you need for this spike (and the focus on a graph to represent the game locations).
- Make a list of the type of details that need to be stored at each location.
- Convert your list into a node design (class?) and a graph design. Identify what graph based functions you will need to move a player around the world.
- To move the player how will you alter the graph? Does the graph contain a reference to the player? Does the Player contain a reference to the graph? What potential advantages/disadvantages are offered by each approach?
- Create a very simple text file. Write code to load the file (maybe just print the details back to screen to start with) and then create a graph using the loaded details.
- Implement the basic (minimum) “go” commands you need to test your graph data.
- Test early, test often.