Task 13   Spike: Composite Pattern

CORE

Context

A text-based adventure game can create an immersive experience where entities of the game can be compositions of other entities. To do this, player commands need to act on "entities" of the game, some of which are composed of other entities.

Knowledge/Skill Gap:

The developer needs to know how to create and modify, for a text-based adventure game, game entities that are composed of other game entities.

Goals

Building on the work of earlier Zorkish Spikes, extend the game world to support game entities composed of other game entities. Update the command parser and command manager for the game to support actions that modify the composition and location of game entities.

Create part of the Zorkish game that demonstrates the following:

1. Adventure (world) files that include the specification of game entities, their properties, and any nested entities (composition) they may contain.
2. Players are able to observe and modify entities (what they contain, and their location) ie. "look in", "take _ [from] _", "put _ in _", "open _ [with] _"

Expected Output

Repository
1. Code
2. Spike Report

Canvas
1. Spike Report

Notes

- Read the game specifications again.
- Research dictionaries – collections that can access contents using string keys. (STL)
- A dictionary collection making reference to objects using strings as keys, and an OO command pattern. The game location graph can be extended to support entities that are collections of entities – this is the essence of the OO composite pattern!
- Put designs and plans on paper. Think as much as possible before you code! (If you do this, be sure to include your paper design with your outcome report.)
- Create a new adventure file that contains a minimal game world description and some entities that also contain other entities that you can use later for testing.
- Update the adventure loading code so that your game world (graph) supports the entities and the composite pattern
- Extend the player commands (the command pattern/manager) to enable modification of game entities composition and test
- Implement other commands – and test… extend… until done.
- Test. Check for memory leaks… (Seriously!)