Task 21 Spike: Control Mapping & Configuration Files

CORE

Context

In games we need to be able to capture control input, from devices such as keyboard, control pads, mice etc. and direct input to the appropriate parts of their game components so that they can respond. It is considered a good architecture, and supports good user experience, to allow "mapping" of input signals (messages) to game components. The alternative of "hard-coding" input response to components is a poor approach. Configuration files allow for more flexible products if implemented correctly. Loading a configuration file instead of using hard-coded values increases game design opportunities.

Knowledge/Skill Gap:

The developer to needs to know how to support user-specified control mapping saveable to file so that the components of a game (such as player characters) can respond to input in a flexible and reconfiguration way.

Goals

Create a simple application (that uses your framework (SDL2) capture input events (such as key down or up events), and map the input to a change.

You must specifically demonstrate:

1. That input can be mapped to components based on configuration data at run-time, and
2. Be able to change and reload-input mapping configuration without restarting the application.

Expected Output

Repository

1. Code
2. Spike Report

Canvas

1. Spike Report

Notes

- A simple console application is fine to use, or your previous Sprites and Graphics work.
- Use a simple text file for the mapping configuration.
- You don't need many commands to prove the concept.
- Keyboard mapping is easy to demonstrate. Mouse settings can get interesting if you want to develop that aspect of your skills.