Task 29 Spike: ECS & Performance

OPTIONAL

Context

An Entity Component System (ECS) is an architectural software pattern often used in games and game frameworks. It is used as an alternative to deep inheritance hierarchy that often don't fit well with complex or diverse game entities, and instead uses composition. Composition provides a flexible way to define game entity data, state or behaviour. The "system" idea, promoted to a first-class feature of the pattern, performs global actions on all components, and is disconnected from the entities.

Knowledge/Skill Gap:

The developer has wants to understand and measure the benefits of an ECS with respect to either memory or performance gains.

Goals

Extend the previous ECS spike to work so that you can explore and demonstrate either the memory benefit of an ECS pattern, or the computational benefits and opportunities of an ECS pattern. Present your result as part of your spike report using a visual chart or table to summaries your outcomes.

You can define your own performance target goal to explore. However, we strongly suggest you get this approved by, or at least discuss this with, your tutor first. You will need to set up your experiment, collect relevant measurements for different configurations (or code implementation changes), and present your results in your spike report. Look back at the gap – close it. Can you show a difference with or within an ECS application?

A common performance idea to consider is how long it takes to perform a set number of component operations. For example, consider a group of 1000 sprite-type characters in a game. How long does it take to update all their positions (using velocity) with an ECS architecture? Similarly, if you are able to divide work across different threads, using an ECS pattern and a system per thread, what is the performance gain? There are also different aspects of the ECS implementation that can impact on performance, such as the data structures used to store components and access systems. If you have that level of control over your code, then you could try different data types of containers and see if you can measure the performance impact.

Expected Output

Repository

1. Code
2. Spike Report

Canvas

1. Spike Report