OOP versus ECS for Games

# Quick Summary

* The Entity-Component-System (ECS) is a programming pattern that breaks down the codebase into three parts: entities, components and systems. The entity is a unique ID, the component is statefulness stored in an array-by-state (e.g. health), and the system is the stateless functionality that transforms the state.
* This is different from the more traditional approach of Object-Orientated-Programming (OOP). OOP uses objects to store state which is then transformed by other objects and methods.

# Advantages

* ECS offers performance benefits over OOP. This is due to its memory efficiency reducing cache misses, for example.
* ECS makes the codebase more modular, as it separates data from behaviour.
* ECS allows the creation of dynamic objects at runtime, as these are not defined.
* ECS avoids the problem of deep inheritance.
* OOP is familiar and simpler to grasp, ensuring quicker uptime for new team members and potentially shortens development time.
* OOP can be debugged more quickly due to the integration of data and functionality.
* OOP more closely adheres to encapsulation as a principle.
* OOP promotes code reusability because of inheritance.

# Project Aims and Implementation

* The aim of the project is to compare the performance of an ECS system with an OOP system when performing CPU tasks.
* I realised during planning that converting an entire game from ECS to OOP would be excessive to prove the point, and so designed a proof of concept instead.
* I have provided stopwatch comparisons of the codebases. ECS is faster.