

# Dissertation Example

Thomas Jacobs

October 2022

# Abstract

Thomas Jacobs: An implementation of the Entity Component System model for real-time systems.

Bachelor of Science Dissertation, ;Pages;

The University of Suffolk.

Degree: Computer Games Programming.

;Information;

# Acknowledgements

Pending.

# Glossary

**Architectural Pattern** The structural arrangement and corporation of a system. 1

**Data Oriented Programming** A programming paradigm that focuses on simplifying the complexity of object oriented programming often by separating data and function. . 1

**Game Engine** A framework designated specifically for game development, often composed of multiple multi-media library's such as graphics, physics and audio libraries. 1

**Object Oriented Programming** A Programming paradigm where software is designed around the utilisation of objects composed of procedures and data . 1

**Programming Language Paradigm** Classifies programming languages based upon their features. . 1

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Background (Literature Review)</b>	<b>6</b>
2.1	What is an Entity Component System? . . . . .	1
2.2	What are the advantages associated with an Entity Component System? . . . . .	1
2.3	What are the practical ways an ECS can be implemented and utilised? . . . . .	2
2.4	Problems associated with the Entity Component System . . . . .	2

# Chapter 1

## Introduction

Ever since the marketisation of computer games and more importantly game engines, programmers have sought out methods for making game engines increasingly more extendable to developers

An Entity Component System is an architectural pattern

## Chapter 2

# Background (Literature Review)

## **Abstract**

The intention of this project is to provide an interpretation for how a programmer can develop and implement the ECS model. Unfortunately, this intention came to fruition due to the lack of academic literature that covers the real implementation and development of an ECS, meaning it was difficult to find literature that was anything other than theoretical but nonetheless the literature breakdown should serve as guide for how the architecture will be constructed.



The desire for conducting a literature review for this project is to define and access current definitions and understanding of the ECS model. By comparing multiple sources, the project can devise an Entity Component System that utilises implementations most beneficial for interactive-based systems.

Finally and more importantly, the research will ensure that the Entity Component System developed for this project remains true to the intent of the original model unless it is unavoidably, beneficial to adjust variable elements.

## 2.1 What is an Entity Component System?

Defining an Entity Component System: Literature Source: (Härkönen, 2019) Advantages and Implementation of Entity Component Systems.

Before Härkönen introduces the implementation of an Entity Component System, a clear set of definitions is established, outlining and distinguishing each vital element of The Entity Component System.

Starting with entities, Härkönen describes an entity as the "fundamental building block" of the system. Härkönen specifies that an entity consists of nothing more than a unique identifier; without data or behaviour (methods). An entity's unique identifier can compose groups of components together, forming the sense of a game object or actor.

Components introduce the flow of data in an Entity Component System. Härkönen defines them as small, generic and reusable types that define the properties of an entity. What components do not provide, however, is any form of behaviour or logic. For example, a transformation data structure could derive from a generic component interface. It would describe the position, rotation and scale of an Entity but not process any physical or mathematical logic/operation.

To process mathematical operations on a transformative component, the Entity Component System utilises Component Systems. Härkönen defines these systems as the elements of an ECS that elicit logic. More importantly, Härkönen mentions that a system is global by design rather than being designated to a specific entity; instead, targeting a group of entities with desired components.

For example, a rendering system could target entities with sprite/renderable components and render them to screen.

## 2.2 What are the advantages associated with an Entity Component System?

Source: <https://trepo.tuni.fi/handle/123456789/27593>

In Toni Harkonen's paper ECS is again compared to the often default, object oriented approach. One immediate benefit Harkonen outlines is how the ECS architecture avoids the deadly diamond problem. The deadly diamond problem relates to how a hierarchical structure of inheritance can lead to derived class that inherits from two or more parent classes that both inherit the same base class. ECS avoids this by avoiding literal inheritance all together.

One other major positive of the ECS, highlighted in the paper relates to how the architecture has a close relationship with data oriented design. Data oriented design often derives architecture that is efficient in how it process data for example, the paper mentions how DOD has the possibility for optimised cache utilization that can be achieved by ordering the data in memory. For an Entity Component System this would likely be having component pools made up of individual component instances that are processed together.

## 2.3 What are the practical ways an ECS can be implemented and utilised?

This paper provides a practical outlook for how Entity Component Systems can be used in the real world of software development. The paper describes the use of an ECS in a graphics drawing application. Contrasting to the first sources discussed on what an entity component system is we can see similar description for the key components.

## 2.4 Problems associated with the Entity Component System

In this paper [[https://link.springer.com/content/pdf/10.1007/978-3-319-07440-5\\_49.pdf](https://link.springer.com/content/pdf/10.1007/978-3-319-07440-5_49.pdf)] > *an important and major issue is highlighted in this paper where an ECS could become inefficient if a system*

# Bibliography

- [1] Toni Härkönen. Advantages and implementation of entity-component-systems.
- [2] Dennis Nilsson and Anton Björkman. Profiling a prototype game engine based on an entity component system in c++.
- [3] Thibault Raffaillac and Stéphane Huot. Polyphony: Programming interfaces and interactions with the entity-component-system model. 3:22.