# Abstract

# List of Figures

# List of Tables

# Chapter 1: Introduction

## Background No research in using Rust for military simulation exists. But Rust, with benefits over other languages, can be used to design a flight dynamics model which can be structured as a Data Oriented Design, specifically using the ECS pattern. The DOD motivates the use of cpu cache in video game development. The ECS pattern organizes code for efficient cache execution and separates components of the games to eliminate ambiguous inheritance hierarchies. This paradigm may be a better option than Object Oriented Programming, which supports clean design and flexibility of components.

## Problem Statement

Explore the creation of Rust based software implementing the ECS pattern (Specs library) to create and interactive flight simulation that communicates with a visualization system.

### Past research has proven that DOD paradigm has the potential to improve the speed, lifespan, and maintainability of RIS software. The only prevalent application of DOD into real systems is seen in the gaming world. The entity-component-system (ECS)framework is becoming a widely used DOD framework for building games. Very few instances of DOD is seen outside of this world. Research must be done to see if the potential of DOD can be realized through working software. This paper will explore the ability to create RIS military software using the DOD paradigm, or more specifically the ECS framework, to see if the tested research can be applied to real systems.

## Research Goals

### The goal of this research is show an executable program able to show a successful simulation of an aircraft flying using the design approach which promotes clean design.

## Hypothesis

## The simulation created using ECS is a better alternative than other approaches, such as OOP.

## Approach

## Assumptions / Limitations

## Contributions

## Thesis Overviews

# Chapter 2: Background Overview

## Data Oriented Design Rust (important features/benefits)

## Comparison against C++ Differences between DOD and OOP

## Entity Component System framework

## Entity Component

### System Execution Pattern Specs Socket Programming implementation

Flight Dynamics Model (big picture math involved in simulation)  
Summary

# Chapter 3: Methodology Showcase the functional program experiments Variables Assumptions

# Chapter 4: Results & Discussion

Demonstrate fidelity of the program experiments

# Chapter 5: Conclusion & Future Work