**System Design Document**

**For**

**Aerial Swarm Simulator**

Team member: Elijah Keck

|  |  |
| --- | --- |
| Version/Author | Date |
| <1.0> Dillon Mead | 13/09/21 |
| <1.0> Elijah Keck | 27/09/21 |
| <2.0> Elijah Keck | 26/10/21 |
|  |  |

TABLE OF cONTENTs

[1 INTRODUCTION 3](#_Toc530500649)

[1.1 Purpose and Scope 3](#_Toc530500650)

[1.2 Project Executive Summary 3](#_Toc530500651)

[1.2.1 System Overview 3](#_Toc530500652)

[1.2.2 Design Constraints 3](#_Toc530500653)

[1.2.3 Future Contingencies 3](#_Toc530500654)

[1.3 Document Organization 3](#_Toc530500655)

[1.4 Project References 4](#_Toc530500656)

[1.5 Glossary 4](#_Toc530500657)

[2 SYSTEM ARCHITECTURE 4](#_Toc530500658)

[2.1 System Hardware Architecture 4](#_Toc530500659)

[2.2 System Software Architecture 4](#_Toc530500660)

[2.3 Internal Communications Architecture 4](#_Toc530500661)

[3 HUMAN-MACHINE INTERFACE 4](#_Toc530500662)

[3.1 Inputs 5](#_Toc530500663)

[3.2 Outputs 5](#_Toc530500664)

[4 DETAILED DESIGN 5](#_Toc530500665)

[4.1 Hardware Detailed Design 6](#_Toc530500666)

[4.2 Software Detailed Design 6](#_Toc530500667)

[4.3 Internal Communications Detailed Design 7](#_Toc530500668)

[5 EXTERNAL INTERFACES 7](#_Toc530500669)

[5.1 Interface Architecture 7](#_Toc530500670)

[5.2 Interface Detailed Design 8](#_Toc530500671)

[6 SYSTEM INTEGRITY CONTROLS 8](#_Toc530500672)

System Design Document

# INTRODUCTION

## Purpose and Scope

The purpose of this System Design Document is to provide design details of the aerial swarm simulator system. This document encompasses system architecture, the human machine interface, the software design, the communication design, external interfaces, and system integrity.

## Project Executive Summary

This section provides an overview of the Aerial Swarm Simulation project from a management perspective, this shows the system design framework.

### System Overview

This system uses Unreal Engine to simulate the environment and drones make up the simulation. The system takes input from the user on what the drone swarm needs to do. The system then simulates the drone swarm flying to and accomplishing the given mission. The user receives data from the simulated drones which appear on the screen. This data is saved for future use.

A system overview use case diagram is available in Appendix A – Figure 1. This diagram shows the actors and how they interact with the system. The swarm, payload, and ground control are all actors on the swarm system.

### Design Constraints

The design constraints are that the system must be able to run on Unreal Engine. The system must also be able to function autonomously with the only human input being the mission details.

### Future Contingencies

We have not thought about this yet.

## Document Organization

This document is organized by section. These sections break into more detailed pieces. The architecture and detailed design break down into hardware, software, and internal communications. The interface sections break down into inputs and outputs, and architecture and design. The final section covers the system integrity.

## Project References

We have not thought of this yet.

## Glossary

There are no abbreviations yet.

# SYSTEM ARCHITECTURE

This section provides an overview of the software system architecture for the Aerial Swarm Simulator.

## System Hardware Architecture

No hardware architecture required. This is strictly a software application.

## System Software Architecture

The software is split into three modules: the swarm, ground control, and the simulation. The swarm module communicates internally to a swarm leader drone that connects to the ground station.

The swarm leader drone is responsible for sending messages to the individual drones and to the ground station. The individual drones communicate their positions to the lead drone and the lead drone communicates back adjustments based on the swarm algorithm. The drones also communicate their sensor data and status which is packaged by the lead drone and sent back to ground control.

The simulation module holds the responsibility for creating and maintaining the environment and drone objects within that environment. The ground control module is responsible for the user interface. This includes the visual portion of the simulation, visualizing drone status, and visualizing the received data from the drones. Ground control is also responsible for sending the mission to the lead drone in the drone swarm.

Figures 2 and 3 in Appendix A detail the data flows through the simulation system. Figures 4 and 5 are the class models that detail the system and subsystem structures of the Aerial Swarm Simulation. Shown in the class models are the breakdown of the composition of the subsystems and how the system connects as a whole.

## Internal Communications Architecture

We have not thought about this yet.

# HUMAN-MACHINE INTERFACE

The human-machine interface involves a user interface that allows the user to visualize the simulation, drone statuses, and the sensor data from the drone payloads. The interface has another important function, sending the mission to the drone swarm. This section details the inputs and outputs of this interface.

## Inputs

The input for this interface is the mission data to send to the drone swarm. The specific types of data are still being worked on.

## Outputs

The outputs for this interface are the visual representation for the drone swarm, the sensor data, and the drone statuses.

# DETAILED DESIGN

This section contains the detailed software design for the Aerial Swarm Simulator.

## Hardware Detailed Design

No hardware utilized.

## Software Detailed Design

The software, as mentioned in section 2.2 has three modules: the swarm, the simulation, and ground control. Using figures 1, 2, and 3 in Appendix A, one can see the use cases and data flow for the system. In Figure 1 the use case diagram shows the actors being the three software modules. These modules communicate through the system to form the swarm, assign the mission, monitor drone status, and transmit the payload data. This diagram shows the interconnectivity of the modules and the relationships between them in accomplishing these use cases. In figure 2, the diagram shows the base level of data flow in the system. The swarm transmits position, mission status, and individual drone statuses through the system. In return the swarm receives mission assignments and position updates. The payload sends the measured data into the system as well. Ground control receives the drone and mission statuses as well as the payload data from the system. Ground control transmits the mission data into the system as well. Figure 3 provides a more in-depth diagram of the inner workings of how the data is transferred through the system. The swarm algorithm process takes in the drone positions and provides the adjustments needed for those drone positions. It also transmits the mission to the drone swarm. The payload data store stores the payload data that is transmitted into the system. The ground control draws the data out of this data store to display to the user. The process status process takes in the drone and mission statuses from the drone swarm and processes and relays that information to be displayed to the user in ground control.

Figures 4 and 5 show the breakdown of the system and different subsystems in a class model. The system is connected with associations between the three subsystems: Data Link, UAV, and Ground. These subsystems are detailed in figure 5. The data link subsystem is made up of Command and control, Payload, and External classes in a parent child relationship. The Ground subsystem is an aggregate of Ground Terminal Data, Ground Control Station, Flight Planning, UA Pilot, Launch Recovery Station, and Mission Monitoring classes. The final subsystem is UAV. It is made up of three more subsystems, Aircraft, Payload, and Internal I/O. The Aircraft subsystem is composed of Airframe, Propulsion, and Avionics classes. The Payload subsystem is made up of Chemical Sensor, Camera, Radar, and Temperature Sensor classes. These all inherit characteristics from the Payload Class. The Internal I/O subsystem is made up of Altimiter, Antennae, Camera, GPS, and Inertial Measurement Unit classes. These subsystems compose the the UAV subsystem. Using these figures one can get an in depth idea of the system modules, the use cases, and the data flow through the system.

## Internal Communications Detailed Design

We have not thought about this yet.

# EXTERNAL INTERFACES

We are not currently using external interfaces.

## Interface Architecture

We are not currently using external interfaces.

## Interface Detailed Design

We are not currently using external interfaces.

# SYSTEM INTEGRITY CONTROLS

We have not thought about this yet.

# APPENDIX A

## Use Case – Figure 1

Diagram

Description automatically generated

## Diagram Description automatically generatedData Flow Diagram Level 0 – Figure 2

## Data Flow Diagram Level 1 – Figure 3

Diagram

Description automatically generated

## System Class Model – Figure 4

Chart, diagram, box and whisker chart

Description automatically generated

## Subsystem Class Models – Figure 5

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated