**System Test Plan**

**For**

**Modeling Unmanned Aerial Swarms Using Unreal Engine and AirSim Simulator**

Team member: Naimah-Joy Chapman, Elijah Keck, Dillon Mead, and John Mueller

|  |  |
| --- | --- |
| Version/Author | Date |
| 1.0/Naimah-Joy Chapman | 10/14/2021 |
|  |  |
|  |  |
|  |  |

Table of Contents

[1. Introduction 2](#_Toc530500519)

[1.1 Purpose 2](#_Toc530500520)

[1.2 Objectives 2](#_Toc530500521)

[2. Functional Scope 2](#_Toc530500522)

[3. Overall Strategy and Approach 2](#_Toc530500523)

[3.1 Testing Strategy 2](#_Toc530500524)

[3.2 System Testing Entrance Criteria 2](#_Toc530500525)

[3.3 Testing Types 2](#_Toc530500526)

[3.4 Suspension Criteria and Resumption Requirements 3](#_Toc530500527)

[4. Execution Plan 3](#_Toc530500528)

[4.1 Execution Plan 3](#_Toc530500529)

[5. Traceability Matrix & Defect Tracking 3](#_Toc530500530)

[5.1 Traceability Matrix 3](#_Toc530500531)

[5.2 Defect Severity Definitions 3](#_Toc530500532)

[6. Environment 4](#_Toc530500533)

[6.1 Environment 4](#_Toc530500534)

[7. Assumptions 4](#_Toc530500535)

[8. Risks and Contingencies 4](#_Toc530500536)

[9. Appendices 4](#_Toc530500537)

# Introduction

## Purpose

This document is a test plan for Modeling Unmanned Aerial Swarm using Unreal Engine and AirSim Simulator System Testing, produced by the System Testing team. It describes the testing strategy and approach to testing the team will use to verify that the application meets the established requirements of the business prior to release.

## Objectives

* Meets the requirements, specifications and the Business rules.
* Supports the intended business functions and achieves the required standards.
* Satisfies the Entrance Criteria for User Acceptance Testing.

# Functional Scope

The Modules in the scope of testing for the Modeling Unmanned Aerial Swarm using Unreal Engine and AirSim Simulator System Testing are mentioned in the document attached in the following path:

# Overall Strategy and Approach

## Testing Strategy

Modeling Unmanned Aerial Swarm using Unreal Engine and AirSim Simulator System Testing will include testing of all functionalities that are in scope (Refer Functional Scope Section) identified. System testing activities will include the testing of new functionalities, modified functionalities, screen level validations, workflows, functionality access, testing of internal & external interfaces.

### Function Testing

**Test Objective:** The applications navigation data entry, processing and retrieval work according to the specific requirements in the SRS

**Technique:** Execute use cases from the use case diagram -> when valid data is given then the corresponding results is given, when invalid data is given then a warning message will show.

**Completion Criteria:** When all use case have been tested and al defects have been mitigated

**Special Consideration:** Access to the Unreal Engine and AirSim simulator and the corresponding Systems Requirement Specification document

### Performance Testing

**Testing Objective:** Ensure algorithm can read, calculate and translate values to the UAV.

**Technique:** Execute the rule-based algorithm, compute values.

**Completion Criteria:** UAV should go to new points

## System Testing Entrance Criteria

In order to start system testing, certain requirement must be met for testing readiness. The readiness can be classified into:

## Testing Types

### Usability Testing

User interface attributes, cosmetic presentation and content will be tested for accuracy and general usability. The goal of Usability Testing is to ensure that the User Interface is comfortable to use and provides the user with consistent and appropriate access and navigation through the functions of the application (e.g., access keys, consistent tab order, readable fonts etc.)

### Functional Testing

The objective of this test is to ensure that each element of the component meets the functional requirements of the business as outlined in the:

* Business / Functional Requirements
* Business rules or conditions
* Other functional documents produced during the course of the project i.e. resolution to issues/change requests/feedback

### System Requirements Specification, Req 1: “The Aerial Swarm Simulator system shall be modelled and simulated in Microsoft’s AirSim Simulator”

System Requirements Specification, Req 2: “Ground station shall have the ability to assign, modify, and re-assign missions to the aerial swarm”

System Requirements Specification, Req 3: “Ground station shall receive reports on the aerial swarm status including status of all individual UAV.”

System Requirements Specification, Req 4: “Ground station shall receive sensor data from aerial swarm.”

System Requirements Specification, Req 5: “Ground station shall record sensor data.”

System Requirements Specification, Req 6: “Ground station shall display aerial swarm status including status of all individual UAV.”

System Requirements Specification, Req 7: “The aerial swarm shall designate a lead UAV for swarm organization and communication.”

System Requirements Specification, Req 8: “The aerial swarm shall reassign the lead UAV when the current lead becomes inactive.”

System Requirements Specification, Req 9: “The lead UAV shall receive status data from all individual UAV.”

System Requirements Specification, Req 10: “The lead UAV shall transmit status data of the aerial swarm and all individual UAV.”

System Requirements Specification, Req 11: “The aerial swarm shall transmit sensor data to a repository in the ground station.”

System Requirements Specification, Req 12: “The aerial swarm shall determine the positioning of individual UAV and transmit the data to individual UAV.”

System Requirements Specification, Req 13: “The aerial swarm shall adjust and continue the mission task when an individual UAV becomes inactive.”

System Requirements Specification, Req 14: “The aerial swarm shall acknowledge receipt, or modification, of mission task.”

System Requirements Specification, Req 15: “The aerial swarm shall return to ground station when mission task is complete.”

System Requirements Specification, Req 16: “Individual UAV shall communicate position and status with the aerial swarm.”

System Requirements Specification, Req 17: “Individual UAV shall avoid collisions with objects including other UAV.”

System Requirements Specification, Req 18: “Individual UAV shall carry a payload that will house sensors”

System Requirements Specification, Req 19: “Sensor data shall be routed through the aerial swarm via the lead UAV.”

System Requirements Specification, Req 23: “The aerial swarm shall continue task and attempt to finish the mission queue when any individual UAV becomes inactive.”

## Suspension Criteria and Resumption Requirements

This section will specify the criteria that will be used to suspend all or a portion of the testing activities on the items associated with this test plan.

### Suspension Criteria

Testing will be suspended if the incidents found will not allow further testing of the system/application under-test. If testing is halted, and changes are made to the hardware, software or database, it is up to the Testing Manager to determine whether the test plan will be re-executed, or part of the plan will be re-executed.

### Resumption Requirements

Resumption of testing will be possible when the functionality that caused the suspension of testing has been retested successfully.

# Execution Plan

## Execution Plan

The execution plan will detail the test cases to be executed. The Execution plan will be put together to ensure that all the requirements are covered. The execution plan will be designed to accommodate some changes if necessary, if testing is incomplete on any day. All the test cases of the projects under test in this release are arranged in a logical order depending upon their inter dependency.

### Database Testing (See 3.1.2)

### Function Testing (See 3.1.1)

### Performance Testing (See 3.1.1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement (From SRS) | Test Case ID | Input | Expected Behavior | Pass/Fail |
| [Req 1] The Aerial Swarm Simulator system shall be modelled and simulated in Microsoft’s AirSim Simulator. | 1.1 | The play scenario button is pressed in Unreal Engine Editor | The simulation loads in the environment, no errors are thrown | Pass |
| [Req 2] Ground station shall have the ability to assign, modify, and re-assign missions to the aerial swarm. | 2.1 | User selects mission path | Aerial swarm acknowledges mission. |  |
| [Req 3] Ground station shall receive reports on the aerial swarm status including status of all individual UAV. | 3.1 | Data set for UAV and Swarm status | Ground Control Station receives data set |  |
| [Req 4] Ground station shall receive sensor data from aerial swarm. | 4.1 | Data set for sensor data | Ground Control Station receives data set |  |
| [Req 5] Ground station shall record sensor data. | 5.1 |  |  |  |
| [Req 6] Ground station shall display aerial swarm status including status of all individual UAV. | 6.1 | Data set of UAV and Aerial Swarm status | Data set displayed in AirSim for User |  |
| [Req 7] The aerial swarm shall designate a lead UAV for swarm organization and communication. | 7.1 | “Lead” attribute changed to False | A single UAV’s “lead” attribute changed to True |  |
| [Req 8] The aerial swarm shall reassign the lead UAV when the current lead becomes inactive. | 8.1 | Lead UAV status changed to “inactive” | Another UAV “lead” attribute changed to True |  |
| [Req 9] The lead UAV shall receive status data from all individual UAV. | 9.1 | Data set from each individual UAV | Lead UAV receives data transmission |  |
| [Req 10] The lead UAV shall transmit status data of the aerial swarm and all individual UAV. | 10.1 | Data set of Swarm and individual UAV status | Ground Control Station receives transmission |  |
| [Req 11] The aerial swarm shall transmit sensor data to a repository in the ground station. | 11.1 | Data set from sensors |  |  |
| [Req 12] The aerial swarm shall determine the positioning of individual UAV and transmit the data to individual UAV. | 12.1 | Aerial Swarm issued formation | Swarm moves into formation position |  |
| [Req 13] The aerial swarm shall adjust and continue the mission task when an individual UAV becomes inactive. | 13.1 |  |  |  |
| [Req 14] The aerial swarm shall acknowledge receipt, or modification, of mission task. | 14.1 |  |  |  |
| [Req 15] The aerial swarm shall return to ground station when mission task is complete. | 15.1 |  |  |  |
| [Req 16] Individual UAV shall communicate position and status with the aerial swarm. | 16.1 |  |  |  |
| [Req 17] Individual UAV shall avoid collisions with objects including other UAV. | 17.1 |  |  |  |
| [Req 18] Individual UAV shall carry a payload that will house sensors. | 18.1 |  |  |  |
| [Req 19] Sensor data shall be routed through the aerial swarm via the lead UAV. | 19.1 |  |  |  |
| [Req 23] The aerial swarm shall continue task and attempt to finish the mission queue when any individual UAV becomes inactive. | 23.1 |  |  |  |

# Traceability Matrix & Defect Tracking

## Traceability Matrix

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Req.** | **Req. Depend.** | **Test Case** | **Test Case Depend.** | **Respon-sible** | **Result** | **Comment** |
| R1 |  | 1.1 |  |  |  |  |
| R2 | R1 | 2.1 | 1.1 |  |  |  |
| R3 | R1 | 3.1 | 1.1 |  |  |  |
| R4 | R1, R3 | 4.1 | 1.1, 3.1 |  |  |  |
| R5 | R1, R3, R4 | 5.1 | 1.1, 3.1, 4.1 |  |  |  |
| R6 | R1, R3 | 6.1 | 1.1, 3.1 |  |  |  |
| R7 | R1, | 7.1 | 1.1 |  |  |  |
| R8 | R1, R7 | 8.1 | 1.1, 7.1 |  |  |  |
| R9 | R1, R3, R7 | 9.1 | 1.1, 3.1, 7.1 |  |  |  |
| R10 | R1, R7, R8 | 10.1 | 1.1, 7.1, 8.1 |  |  |  |
| R11 | R1, R4, R5 | 11.1 | 1.1, 4.1, 5.1 |  |  |  |
| R12 | R1 | 12.1 | 1.1 |  |  |  |
| R13 | R1, R12 | 13.1 | 1.1, 12.1 |  |  |  |
| R14 | R1, R2 | 14.1 | 1.1, 2.1 |  |  |  |
| R15 | R1, R2 | 15.1 | 1.1, 2.1 |  |  |  |
| R16 | R1 | 16.1 | 1.1 |  |  |  |
| R17 | R1 | 17.1 | 1.1 |  |  |  |
| R18 | R1 | 18.1 | 1.1 |  |  |  |
| R19 | R1, R7, R8, R18 | 19.1 | 1.1, 7.1, 8.1, 18.1 |  |  |  |
| R20 | R1, R2 | 20.1 | 1.1, 2.1 |  |  |  |
| R21 | R1, R2 | 21.1 | 1.1, 2.1 |  |  |  |
| R22 | R1, R2 | 22.1 | 1.1, 2.1 |  |  |  |
| R23 | R1, R2, R13 | 23.1 | 1.1, 2.1, 13.1 |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

## Defect Severity Definitions

|  |  |
| --- | --- |
| **Critical** | The defect causes a catastrophic or severe error that results in major problems and the functionality rendered is unavailable to the user. A manual procedure cannot be either implemented or a high effort is required to remedy the defect. Examples of a critical defect are as follows:   * System abends * Data cannot flow through a business function/lifecycle * Data is corrupted or cannot post to the database |
| **Medium** | The defect does not seriously impair system function can be categorized as a medium Defect. A manual procedure requiring medium effort can be implemented to remedy the defect. Examples of a medium defect are as follows:   * Form navigation is incorrect * Field labels are not consistent with global terminology |
| **Low** | The defect is cosmetic or has little to no impact on system functionality. A manual procedure requiring low effort can be implemented to remedy the defect. Examples of a low defect are as follows:   * Repositioning of fields on screens * Text font on reports is incorrect |

# Environment

## Environment

* The System Testing Environment will be used for System Testing.

# Assumptions

• Assumed no malicious actors

• Assumed AirSim running without error

• Assumed Unreal running without error

# Risks and Contingencies

One risk is data corruption. As the simulation can take pictures and data measurements during the simulation and save them to the host computer, there could be risk of losing this data in the event of an unexpected shutdown of the simulation. The contingency for this risk is to save the data to a more permanent location once received by ground station during the simulation.

# Appendices

# 