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
1.1.0/Naimah-Joy Chapman	10/14/2021
1.1.1/John Mueller	10/18/2021
1.1.2/Elijah Keck	10/19/2021
1.1.2/Dillon Mead	10/19/2021
1.2.0/Naimah-Joy Chapman	12/02/2021
2.1.0/Elijah Keck	02/22/2022
2.1.1/Dillon Mead	02/27/2022

Table of Contents

1. Introduction

1.1 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 system meets the established requirements of the business prior to release.

1.2 Objectives

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

1.3 Document Conventions

Work accomplished during the Spring semester is located below the identification of "**2nd Semester**". Text highlighted in yellow are portions of the product not currently developed. These features will be explored during future development.

2. 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:

- The System Requirements Specification document: https://github.com/mead-d/Modeling-Unmanned-Aerial-Swarms-Using-Unreal-Game-Engine-and-AirSim-Simulator/blob/d865354b244f6d549a49221854d58130624ee19d/SRS/Graded%20System%20Requirements%20Specification.doc

3. Overall Strategy and Approach

3.1 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.

3.1.1 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 cases have been tested and all defects have been mitigated
Special Consideration: Access to the Unreal Engine and AirSim simulator and the corresponding Systems Requirement Specification document

3.1.2 Performance Testing

Testing Objective: Ensure algorithm can read, calculate and translate values (distance,

area) to the UAV.
Technique: Execute the rule-based algorithm, compute values.
Completion Criteria: UAV should go to new points

3.2 System Testing Entrance Criteria

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

3.3 Testing Types

3.3.1 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.)

3.3.2 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:

- Functional Requirements
- Other functional documents produced during 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: "Visual Studio 2019 shall be used to edit files and environment variables."

System Requirements Specification, Req 3: "The user shall implement mission scenarios by executing the appropriate script in the AirSim simulator"

System Requirements Specification, Req 4: "System data shall be display to the user through the python environment terminal."

System Requirements Specification, Req 5: "Ground station shall have the ability to assign to the aerial swarm."

System Requirements Specification, Req 6: "Ground station shall receive reports on the aerial swarm status including status of all individual UAV."

System Requirements Specification, Req 7: "Ground station shall receive sensor data from aerial swarm."

System Requirements Specification, Req 8: "Ground station shall record sensor data."

System Requirements Specification, Req 9: "Ground station shall display aerial swarm status including status of all individual UAV."

System Requirements Specification, Req 10: "The aerial swarm shall designate a lead UAV for swarm organization and communication."

System Requirements Specification, Req 11: "The aerial swarm shall reassign the lead

UAV when the current lead becomes inactive."

System Requirements Specification, Req 12: "The aerial swarm shall have at least three UAV for any given mission"

System Requirements Specification, Req 13: "The lead UAV shall receive status data from all individual UAV every 0.1 seconds."

System Requirements Specification, Req 14: "The lead UAV shall transmit status data of the aerial swarm and all individual UAV."

System Requirements Specification, Req 15: "The aerial swarm shall measure the volume of an identified object."

System Requirements Specification, Req 16: "The aerial swarm shall transmit sensor data to a repository in the ground station."

System Requirements Specification, Req 17: "The aerial swarm shall determine the positioning of individual UAV and transmit the data to individual UAV."

System Requirements Specification, Req 18: "The aerial swarm shall adjust and continue the mission task when an individual UAV becomes inactive."

System Requirements Specification, Req 19: "The aerial swarm shall return to "home" location when mission task is complete."

System Requirements Specification, Req 20: "Individual UAV shall communicate position with the aerial swarm using North, East, and Down coordinates."

System Requirements Specification, Req 21: "Individual UAV shall communicate active status with the aerial swarm."

System Requirements Specification, Req 22: "Individual UAV shall avoid collisions with objects including other UAV."

System Requirements Specification, Req 23: "Individual UAV shall carry a payload that will house sensors."

System Requirements Specification, Req 24: "Sensor data shall be routed through the aerial swarm via the lead UAV."

2nd Semester

System Requirements Specification, Req 25: "A UAV shall avoid collisions with all other objects while moving."

System Requirements Specification, Req 26: "The aerial swarm shall avoid collisions with all other objects."

System Requirements Specification, Req 27: "The aerial swarm shall have the ability to sense objects in the environment."

System Requirements Specification, Req 28: "The Collision Avoidance system shall continuously execute."

System Requirements Specification, Req 29: "The Collision Avoidance system shall identify an impending collision."

System Requirements Specification, Req 30: "The Collision Avoidance system shall implement the avoidance algorithm when an impending collision is detected."

System Requirements Specification, Req 31: "The Collision Avoidance system shall not implement the avoidance algorithm if an impending collision is not detected."

System Requirements Specification, Req 32: "The UAV or aerial swarm shall continue on the original mission path after successfully avoiding a collision."

System Requirements Specification, Req 33: "The avoidance algorithm shall choose an avoidance path without additional impending collisions."

System Requirements Specification, Req 34: "The Collision Avoidance system shall record each impending collision."

System Requirements Specification, Req 35: "The Collision Avoidance system shall record every execution of the avoidance algorithm."

System Requirements Specification, Req 36: "The Collision Avoidance system shall record any collision between the UAV and another object."

1st Semester

System Requirements Specification, Req 37: "The aerial swarm shall continue task and attempt to finish the mission queue when any individual UAV becomes inactive."

System Requirements Specification, Req 38: "The aerial swarm shall report to ground control when a UAV becomes inactive."

3.4 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.

3.4.1 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.

3.4.2 Resumption Requirements

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

4. Execution Plan

4.1 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.

4.1.1 **Function Testing** (See 3.1.1)

4.1.2 **Performance Testing** (See 3.1.2)

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] Visual Studio 2019 shall be used to edit files and environment variables.	2.1	Visual Studio 2019 is opened	Files and environment variables are edited in Visual Studio 2019	Pass
[Req 3] The user shall implement mission scenarios by executing the appropriate script in the AirSim simulator	3.1	Execute the appropriate script in AirSim simulator	Mission executes from the appropriate script in AirSim simulator	Pass
[Req 8] Ground station shall receive sensor data from aerial swarm.	8.1	Data set for sensor data	Ground Control Station receives data set	Pass
[Req 10] Ground station shall display aerial swarm status including status of all individual UAV.	10.1	Data set of UAV and Aerial Swarm status	Data set displayed in AirSim for User	Pass
[Req 13] The aerial swarm must have at least three UAV for any given mission	13.1	Mission is assigned three UAVs	Mission always has three UAVs	Pass
[Req 14] The lead UAV shall receive status data from all individual UAV.	14.1	Data set from each individual UAV	Lead UAV receives data transmission	Pass
[Req 15] The lead UAV shall transmit status data of the aerial swarm and all individual UAV.	15.1	Data set of Swarm and individual UAV status	Ground Control Station receives transmission	Pass
[Req 16] The aerial swarm shall measure the volume of an identified object.	16.1	Aerial swarm is given a mission to measure volume of an object	The aerial swarm measures the volume of an identified object	Fail
[Req 18] The aerial swarm shall determine	18.1	Aerial Swarm issued	Swarm moves into formation position	Pass

the positioning of individual UAV and transmit the data to individual UAV.		formation		
[Req 19] The aerial swarm shall adjust and continue the mission task when an individual UAV becomes inactive.	19.1	UAV becomes inactive	Aerial swarm continues mission	Pass
[Req 22] The aerial swarm shall return to ground station when mission task is complete.	22.1	All mission tasks have been completed	Aerial swarm returns to ground station	Pass
[Req 23] Individual UAV shall communicate position with the aerial swarm.	23.1	Individual UAV sends their position to another UAV	Individual UAVs conveys their position between each other	Pass
2nd Semester				
[Req 26] Individual UAV shall carry a payload that will house sensors.	26.1	Payload with a sensor is added to individual UAV	Individual UAV carries a payload that has a sensor	Pass
[Req 27] Sensor data shall be routed through the aerial swarm via the lead UAV.	27.1	Sensor data is collected	Sensor data is routed through the aerial swarm via the lead UAV	Pass
[Req 29] The aerial swarm shall report to ground control when a UAV becomes inactive.	29.1	An individual UAV becomes inactive	The aerial swarm reports back to ground control once an individual UAV becomes in active	Pass
[Req 30] The Collision Avoidance system shall implement the avoidance algorithm when an impending collision is detected.	30.1	Collision detection module returns true for detecting an object within standoff distance	The UAV detecting a possible collision makes a right turn to avoid an object	Pass
[Req 31] The Collision Avoidance system shall not implement the avoidance algorithm if an impending collision is not detected.	31.1	Collision detection module returns false	The UAV continues on the path to the selected waypoint	Pass
[Req 32] The UAV or aerial swarm shall continue on the original mission path after successfully avoiding a	32.1	Collision detection returns false after returning true, avoidance	The UAV resumes movement on a new path to the selected waypoint	Pass

collision.		algorithm finishes execution		
[Req 33] The avoidance algorithm shall choose an avoidance path without additional impending collisions.	33.1	The system gets the velocity from the multirotor state	The UAV performs a right turn in the correct direction based on its direction of movement	Pass
[Req 34] The Collision Avoidance system shall record each impending collision.	34.1	A possible collision is detected	A collision warning is output to the console and logged	Pass
[Req 35] The Collision Avoidance system shall record every execution of the avoidance algorithm.	35.1	A possible collision is detected	An avoidance execution is logged	Fail
[Req 36] The Collision Avoidance system shall record any collision between the UAV and another object.	36.1	A collision is detected	The collision is logged	Fail
1st Semester				
[Req 37] The aerial swarm shall continue task and attempt to finish the mission queue when any individual UAV becomes inactive.	37.1	An individual UAV is deactivated	The swarm reforms into the correct geometry for the number of active drones and continues to the waypoint.	Pass
[Req 38] The aerial swarm shall report to ground control when a UAV becomes inactive.	38.1	An individual UAV is deactivated	A message is logged to console denoting the UAV that failed	Pass

5. Traceability Matrix & Defect Tracking

5.1 Traceability Matrix

Req.	**Req. Depend.**	**Test Case**	**Test Case Depend.**	**Responsible**	**Result**	**Comment**
R1		1.1		Elijah	Pass	
R2	R1 R2	2.1	1.1 2.1	Elijah	Pass	
R3	R1 R2	3.1	1.1 2.1	Elijah	Pass	
R7	R1 R2	7.1	1.1 2.1	Dillon	Pass	

R8	R1 R2 R7	8.1	1.1 2.1 7.1	Dillon	Pass	
R10	R1 R2 R7 R8	10.1	1.1 2.1 7.1 8.1	Dillon	Pass	
R13	R1 R2 R3	13.1	1.1 2.1 3.1	Elijah	Pass	
R14	R1 R2	14.1	1.1 2.1	Elijah	Pass	
R15	R1 R2 R14	15.1	1.1 2.1 14.1	Elijah	Pass	
R16	R1 R2 R3	16.1	1.1 2.1 3.1	Dillon	Fail	
R18	R1 R2	18.1	1.1 2.1	Dillon	Pass	
R19	R1 R2 R13 R14 R18	19.1	1.1 2.1 13.1 14.1 18.1	Dillon	Pass	
R22	R1 R2 R13 R19	22.1	1.1 2.1 13.1 19.1	Elijah	Pass	
R23	R1 R2 R15	23.1	1.1 2.1 15.1	Elijah	Pass	
2nd Semester						
R26	R1 R2 R3	26.1	1.1 2.1 3.1	Dillon	Pass	
R27	R1 R2 R3	27.1	1.1 2.1 3.1	John	Pass	
R29	R1 R2 R3 R27	29.1	1.1 2.1 3.1 27.1	Naimah	Pass	
R30	R1 R2 R3 R29	30.1	1.1 2.1 3.1 29.1	Naimah	Pass	
R31	R1 R2	31.1	1.1 2.1	Naimah	Pass	

	R3 R29		3.1 29.1			
R32	R1 R2 R3	32.1	1.1 2.1 3.1	Elijah	Pass	
R33	R1 R2 R3 R27	33.1	1.1 2.1 3.1 27.1	Dillon	Pass	
R34	R1 R2 R3	34.1	1.1 2.1 3.1	Dillon	Pass	
R35	R1 R2 R3	35.1	1.1 2.1 3.1	Dillon	Fail	
R36	R1 R2 R3	36.1	1.1 2.1 3.1	Dillon	Fail	
1st Semester						
R37	R1 R2 R14 R15	16.1	1.1 2.1 14.1 15.1	Dillon	Pass	
R29	R1 R2 R14 R15 R19	17.1	1.1 2.1 14.1 15.1 19.1	Dillon	Pass	

5.2 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: • Data is corrupted or cannot post to the database • Swarm operation failure • Object recognition failure
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: • User interface displays incorrect data • Drone position data is inaccurate • Object position data is inaccurate

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 • Objects are the wrong color • Text font on data log is incorrect

6. Environment

6.1 Environment

- The System Testing Environment shall be used for System Testing. In order to conduct the testing, the tester needs to have the following installed onto their computer.
  - Unreal Engine v4.27
  - Microsoft's AirSim v1.6.0

7. Assumptions

- Assumed no malicious actors (hackers, viruses, drone pilots)
- Assumed AirSim is correctly configured and running without error
- Assumed Unreal Engine is correctly configured and running without error

8. Risks and Contingencies

Risk #	Risk	Impact	Contingency Plan
1	Data corruption	High	Save the data to the computer's hard drive once received by ground station during the simulation.
2	Shutdown of simulation	High	Restart the program.

9. Appendices

9.1 Glossary

UAV: Unmanned Aerial Vehicle.
UAS: Unmanned Aerial System – "aerial swarm".
Active: UAV is operational and in flight.
Inactive: UAV in not operation or not in flight.