Software Requirements Specification

for

AerialAce

Version 1.0 approved

Prepared by Cinematography Group

University of Alabama at Birmingham

February 23rd, 2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The AerialAce project aims to use an LED-equipped programmable drone to produce visual art in the darkness of the night. Through Python programming, users may create square,rectangular or any shapes that are visible from the ground by directing the drone's flight path. The project's goal is to create visually stunning light displays that viewers below may enjoy by fusing creativity, technology, and aerial photography.

1.2 Product Scope

The AerialAce product scope describes the features and limitations of the software designed to make it easier to create visually stunning displays with drones and LED lights. Under this scope, users will have the ability to coordinate drone flight patterns to create geometric designs that will be visible against the night sky, such as squares, rectangles, or any desirable shape. Customizing the shape's size, adjusting the LED light's brightness and color, and controlling the A drone's fly path using an easy-to-use interface are all considered essential features. It's important to remember that the project does not entail the construction of hardware or complex image processing skills, even though safety precautions are included to guarantee safe operation.
In order to effectively use the software, users must follow the regulations guiding drone operations and have a basic understanding of Python programming.

1.3 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2. Overall Description

2.1 Product Perspective

AerialAce is an innovative software that was developed by enthusiasts who wanted to combine technology and creativity to sketch their creativity on the night sky. It is an independent creative hub that provides users with the means to control drones and arrange light shows. This is an innovative step into the field of aerial art, not just another product in the portfolio. Although it functions autonomously, its smooth integration with current drone and LED light systems guarantees user compatibility. Safety is the top priority, and elements that assist users in following rules and laws regarding drone operation have been included in. When considering AerialAce in the context of aerial artistry, it may be compared to a musician arranging a composition of light in the sky.

2.2 Product Functions

- Drone Control: Give users the ability to control the drone's flying path, including rising, lowering, and forming specific shapes with its movements.
- Shape Configuration: Give customers the ability to specify the sizes and designs of the shapes that will be made in the sky.
- User Interface: Provide a user-friendly interface that enables users to set up drone actions and track the status of light displays.
- Integration: To allow smooth operation, make sure that the drone and LED light systems are compatible with current drone models, so later on, if we want to change the IOT, it should be easy for us.
- Safety Features: Take safety precautions, such as low battery alerts and emergency stop capability.
- Regulatory Compliance: To guarantee legal and safe use, help users follow the rules regarding drone operations and light displays.

2.3 User Classes and Characteristics

2.3.1 User Classes

1. **Drone Enthusiasts:**
   These people love experimenting with different flight patterns and light displays since they have an interest in drones. Their skill levels may vary, going from beginner drone operators to professionals.
2. **Artists and Creatives:**
   This is a group of people who are creative and who want to experiment with nontraditional forms of art. They may not be skilled at operating drones, but they have a good sense of art and are motivated by creative ideas.
3. **Educational Institutions:**
   This user class includes colleges, universities, and schools that utilize AerialAce for educational purposes. The software is used by both teachers and students to explore the fields of programming, technology, and art.
4. **Event Organizers:**
   Event organizers and designers use AerialAce to highlight outdoor parties and events with a little show. Their goal is to leave guests with unforgettable impressions by putting up amazing shows.

2.3.2 Characteristics

1. **Technical Expertise:**
   Users' technical skill levels might differ, ranging from beginners to experts, which affects how they use the program.
2. **Frequency of Use:**
   While educational institutions and event organizers may use the program on occasion for particular events, drone enthusiasts and artists may use it often for their own projects.
3. **Educational Goals:**
   Institutions of higher learning focus on using the software to achieve educational goals and promote research and learning.
4. **Event Planning:**
   Event organizers need to be flexible since they depend on the software to personalize light displays for different kinds of events.

2.4 Operating Environment

The software will be operating on Mac OSX Sonoma 14.3 using a python 3 based program.

2.5 Design and Implementation Constraints

The biggest limitations and constraints include internet speed and latency between the user interface and drone camera/controls. This problem will be overcome through using personal mobile hotspots during execution to minimize latency.

2.6 User Documentation

The main user manual being used is the Python SDK’s README on github.

2.7 Assumptions and Dependencies

The only third-party component being used is the LED light. Code-based third-party components include functions and classes from the SDK.

3. External Interface Requirements

3.1 User Interfaces

The user interface will be built in Python and will consist of basic functions for the different shapes and colors. These user-controlled functions include: selecting a shape, selecting a color, and various premade paths. All of the paths will be preset and hardcoded. The user will also be able to receive a real time video from the drone's camera.

3.2 Hardware Interfaces

The only non-drone based hardware will be the LED light. The LED light will be attached to the bottom of the drone.
s may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3 Software Interfaces

There will be two main software interfaces. The primary is the logic which deals with the connection to the SDK and drone actions. This is where the drone's movements and functions will be contained. The secondary is the GUI which will allow the user to communicate with the drone via a simplified interface. This is where the buttons and real time camera view will be.

3.4 Communications Interfaces

The drone, when on, will create a Wifi access point. When the user's device is connected to this access point, The drone will listen on the network connection for commands sent via the User Datagram Protocol (UDP). The SDK facilitates this connection and makes it easier.

4. System Features

4.1 System Feature 1

4.1.1 Description and Priority

1. The main system features will be divided into 2 separate components: frontend and backend. The frontend will consist of the GUI, the backend will consist of the business logic that communicates with the SDK.
2. The components for each of the layers will include:
   Frontend: buttons (led light control and drone movements), live video feed,
   Backend: pre-coded drone movements, controls for light

Priority (top being most -> to least):

1. Pre-coded drone movements
2. Controls for light
3. Live video feed (least important)

4.1.2 Stimulus/Response Sequences

1. User selects movement button (frontend) -> Drone performs movement (backend)
2. User selects color (frontend) -> LED light changes (backend)

4.1.3 Functional Requirements

REQ-1: The system must allow the user to control drone movements through the frontend interface. Error handling for failed execution will prompt error messages.

REQ-2: The system must enable users to change LED light colors through the frontend with real-time light color changes implemented in the backend. Error handling for failed execution will prompt error messages.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

General Low Latency in controlling the drone is expected: nearing a response of less than 200ms. Maintaining a connection to the drone, within 10 feet, is also necessary.

5.2 Safety Requirements

If latency is too high or connection is lost, drone-related hardware could be damaged. The drone features replaceable propeller blades. In the event that a propeller blade is damaged, a replacement can be provided. To avoid legal complications, avoid using the drone outdoors. The drone must be inspected before each session of use.

5.3 Software Quality Attributes

It is imperative to emphasize the need for the software to possess high adaptability. This entails the capability to seamlessly integrate new features and functionalities with ease of implementation. The program should simplify any advanced concepts within cinematography, such as keyframing (if applicable). By simplifying advanced concepts, the program becomes much easier to use and market.

5.4 Business Rules

All members of the group, Ali Kesserwani, Moby Ashouri, Shouzab Khan, Eric Lim, Maymunah Hicks, and Muhammad Sheri, have unlimited access to controlling and developing the drone.

6. Other Requirements

There are no other requirements.

Appendix A: Glossary

SDK - Tello's Python *Software Development Kit*

LED - light emitting diode that can emit different colors. This light hardware will be attached to the bottom of the drone via an adhesive strip.

Appendix B: To Be Determined List

There are no TBD references.