**TABLE OF CONTENTS**

**1** **Introduction 2**

***1.1*** ***Document overview 2***

***1.2*** ***Abbreviations and Glossary 2***

1.2.1 Abbreviations 2

1.2.2 Glossary 2

***1.3*** ***References 2***

1.3.1 Project References 2

1.3.2 Standard and regulatory References 2

***1.4*** ***Conventions 2***

**2** **Overview of Tests Results 3**

***2.1*** ***Tests log 3***

***2.2*** ***Rationale for decision 3***

***2.3*** ***Overall assessment of tests 3***

***2.4*** ***Impact of test environment 4***

**3** **Detailed Tests Results 5**

***3.1*** ***Sub section name 5***

# **Introduction**

## ***Document overview***

This document is the software test report of the Alpha testing phase of the Aerogotchi software development project. It contains the results of tests, which were executed during the testing phase Alpha.

## ***Abbreviations and Glossary***

### **Abbreviations**

N/A

### **Glossary**

N/A

## ***References***

### **Project References**

| # | Document Identifier | Document Title |
| --- | --- | --- |
| R1 | SRS | Aerogotchi - Software Requirements Specification: Rev C   * OoeyGUI |
| R2 | SDD | Aerogotchi - Software Design Document Rev A   * OoeyGUI |
| R3 | STP | Aerogotchi - Software Test Plan Rev A   * OoeyGUI |

### **Standard and regulatory References**

| # | Document Identifier | Document Title |
| --- | --- | --- |
| N/A |  | N/A |

## ***Conventions***

N/A

# **Overview of Tests Results**

## ***Tests log***

Give a few information about tests.

The AeroGotchi software (version 1.0.0) was tested on the IntelliJ test platform on Windows 11 on 2024/04/23 to 2024/05/07. The tests of the test phase (R3. STP) were executed.

Testers were:

* Keith Chua

## ***Rationale for decision***

After executing a test, the decision is defined according to the following rules:

* **OK:** The test sheet is set to "OK" state when all steps are in "OK" state. The real result is compliant to the expected result.
* **NOK:** The test sheet is set to "NOK" state when all steps of the test are set to "NOK" state or when the result of a step differs from the expected result.

Test results are listed in §3.

## ***Overall assessment of tests***

Give a qualitative overall assessment of tests.

* Testing applications on multiple platforms for Android emulator (old and new) were fully functional.
* Testing application on actual android divide was fully functional.
* Testing performed as intended for application use.
* Testing failed on 1 occurrence via emulator for database post request.

Give quantitative results.

Statistics about tests:

* 100% of tests OK,

## ***Impact of test environment***

Testing on the emulator and actual device proved that the application is fully functional and working as expected. Multiple test cases were performed to verify integrity of application. The difference while using the test software on IDE was it had trouble with doing a post request towards our database where it worked fine on the emulator and actual device. Different testing environments provide different results and we prefer to test it on the emulator and actual device due to integrity of the application.

# **Detailed Tests Results**

For each executed test, this document contains:

* Test identification;
* Test title;
* Test decision;
* A comment containing additional information or problems encountered during execution and differences with the test procedure.

For the problems leading to a bug, the bug ID is reported in the result of the step where problem was encountered.

For each NOK in a step, at least one bug shall be created (or referenced if already exists).

***3.1 Title Screen***

| **Test ID** | **TC-1** | **Comment** | **Decision** |
| --- | --- | --- | --- |
| Test description | Can see TitleScreen |  | OK |
| Verified Requirement | UIR\_SRS\_001 |  |  |
| Initial conditions | Aerogotchi will have a TitleScreen that shows Aerogotchi design | Not relevant to testing |  |
| Tests inputs | N/A |  |  |
| Data collection actions | N/A |  |  |
| Tests outputs | Navigate to next page route (LoginScreen) |  | OK |
| Assumptions and constraints | N/A |  |  |
| Expected results and criteria | Tap on Screen to go to next page route |  | OK |
| **Test procedure** | IntelliJ integration test.  Emulator / Android Device. |  |  |
| **Step number** | **Operator actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Start Application | Show TitleScreen  Should be able to see Aerogotchi Logo | OK |
| 2 | Tap on TitleScreen | Navigate to LoginScreen | OK |

***3.2 Character Screen***

| **Test ID** | **TC-2** | **Comment** | **Decision** |
| --- | --- | --- | --- |
| Test description | First time user login, prompt user for pet name. |  | OK |
| Verified Requirement | UIR\_SRS\_001.2 |  |  |
| Initial conditions | After logging in with email and password |  |  |
| Tests inputs | “Enter pet name” |  | OK |
| Data collection actions | N/A |  |  |
| Tests outputs | N/A |  |  |
| Assumptions and constraints | N/A |  |  |
| Expected results and criteria | Navigate to Pet view screen after entering pet name, |  | OK |
| **Test procedure** | IntelliJ integration test. Emulator / Android Device |  |  |
| **Step number** | **Operator actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Login with email and password | Navigate to Pet Name | OK |
| 2 | Enter Pet name | Navigate to Pet view | OK |
|  |  |  |  |

***3.3 Status Menu***

| **Test ID** | **TC-3** |  | **Decision** |
| --- | --- | --- | --- |
| Test description | Display proper values for Energy, Happiness and Hunger |  | OK |
| Verified Requirement | UIR\_SRS\_006 |  |  |
| Initial conditions | Must be in Pet view |  |  |
| Tests inputs | Update levels and get data from database |  | OK |
| Data collection actions | N/A |  |  |
| Tests outputs | Updates levels |  | OK |
| Assumptions and constraints | Constraints include levels only has a maximum value of 10 |  |  |
| Expected results and criteria | Energy levels are updated with colors. Red, Yellow and Green |  | OK |
| **Test procedure** |  |  |  |
| **Step number** | **Operator actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Navigate to Status Menu | See current status | OK |
| 2 | Feed drone | Update status levels | OK |

***4.1 Face Tracking***

| **Test ID** | **TC-4** | **Comment** | **Decision** |
| --- | --- | --- | --- |
| Test description | Face Tracking is functioning properly and not crashing |  | OK |
| Verified Requirement | FUNC\_SRS\_001.2 |  |  |
| Initial conditions | Must be connected to Drone wifi |  |  |
| Tests inputs | Run python script |  | OK |
| Data collection actions | N/A |  |  |
| Tests outputs | Tracks face and follows accordingly |  | OK |
| Assumptions and constraints | Must be using Tello SDK library |  |  |
| Expected results and criteria | Drone finds face and follows it by moving front/back and up/down |  | OK |
| **Test procedure** |  |  |  |
| **Step number** | **Operator actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Run Python Script | Run program without error | OK |
| 2 | Wait for output window and have face found by drone | Look to get in front of the drone camera so it can form a rectangle on face | OK |
| 3 | Test different methods it can do, such as moving front/back/up/down | Make sure drone is following the face and performing methods as expected | OK |

***4.2 Keyboard Control***

| **Test ID** | **TC-5** | **Comment** | **Decision** |
| --- | --- | --- | --- |
| Test description | Keyboard Control inputs are being read properly |  | OK |
| Verified Requirement | FUNC\_SRS\_001.5 |  |  |
| Initial conditions | Must be connected to Drone wifi |  |  |
| Tests inputs | Run python script |  | OK |
| Data collection actions | N/A |  |  |
| Tests outputs | Keyboard Control with specific keys |  | OK |
| Assumptions and constraints | Must be using Tello SDK library |  |  |
| Expected results and criteria | Drone will be controlled using keys on the keyboard |  | OK |
| **Test procedure** |  |  |  |
| **Step number** | **Operator actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Run python script | Make sure there is no error when opening up the pygame window | OK |
| 2 | Test key inputs to make sure drone is performing and moving how it is expected | Drone moves accordingly depending on what key is pressed. Takeoff and land are also working. | OK |

***4.3 Sky Shuffle***

| **Test ID** | **TC-6** | **Comment** | **Decision** |
| --- | --- | --- | --- |
| Test description | Sky Shuffle dances are all function properly and have checks to be able to run at a certain battery percentage |  | OK |
| Verified Requirement | UIR\_SRS\_008.1 |  |  |
| Initial conditions | Must be connected to Drone wifi |  |  |
| Tests inputs | Run python script |  | OK |
| Data collection actions | N/A |  |  |
| Tests outputs | Three different dances |  | OK |
| Assumptions and constraints | Must be using Tello SDK library |  |  |
| Expected results and criteria | Perform one of three (choose randomly) dances created in the function |  | OK |
| **Test procedure** |  |  |  |
| **Step number** | **Operator actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Run python script | There is no error when running the script. | OK |
| 2 | Perform one of the dance functions by random | Dance function is performed without error and lands when done | OK |

***4.4 Photo Pilot***

| **Test ID** | **TC-7** | **Comment** | **Decision** |
| --- | --- | --- | --- |
| Test description | Photo Pilot takes a picture of the current drone camera position when a specific key is pressed |  | OK |
| Verified Requirement | UIR\_SRS\_008.1 |  |  |
| Initial conditions | Must be connected to Drone wifi |  |  |
| Tests inputs | Run python script |  | OK |
| Data collection actions | N/A |  |  |
| Tests outputs | Takes picture when key is pressed |  | OK |
| Assumptions and constraints | Must be using Tello SDK library |  |  |
| Expected results and criteria | Take picture when key is pressed |  | OK |
| **Test procedure** |  |  |  |
| **Step number** | **Operator actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Run python script | There is no error when running this script | OK |
| 2 | Script includes keyboard control which allows user to control the drone in order to get ready to take a picture | Keyboard control function is running as expected and provides no errors. Output window also opens in order to screenshot | OK |
| 3 | S key is pressed to take screenshot | When S is pressed, a screenshot is taken and the output window freezes for 5 seconds, providing indication that a screenshot was taken. | OK |