RAMones Co.

Disgruntled Avians Project

(DAP)

Prepared by:

Stuart Sessions

Bryan Moreno

Dan Phuong

Test Plan Document

Release Date September 12, 2022

Executive Summary

The Disgruntled Avians Project is an interactive game modeled after the popular Angry Birds™ game on mobile app stores. Due to DAP's minimalistic interpretation of physics, DAP will not have demanding hardware requirements, thus allowing it to run on a variety of desktop hardware platforms. Coding this in Java will further expand this cross-platform availability. Future functionality will include an expansion into the mobile market, micro-transactions, player vs. player game modes, and some of these updates will focus specifically on the physics modeling of DAP.

This document provides technical information regarding testing/verification of DAP.

Document Versioning

Date	Owner	Comment
09/12/22	Dan Phuong	Draft Documentation Creation, Executive Summary, Project Description, Testing Matrix

Project Description

The Disgruntled Avian Project (DAP) is meant to be an interactive video game similar to other projectile motion games such as Angry Birds. RAMones are looking to build the framework for a fun, expandable game that can in the future be sold to bigger gaming companies.

DAP will not require internet connectivity, and will run locally on the computer hardware that it is installed in. DAP will not have steep hardware requirements and will be able to run on a variety of operating systems (ex. Windows or Mac).

We plan to bring a vibrant look and feel to our game application. DAP aims to be easy and intuitive to play. This approachable game design will make it easy to expand into different demographics and have wide appeal with ample range for merchandising opportunities.

Interacting with the DAP GUI should be reminiscent of its more popular counterpart Angry Birds™, as a similar way of interacting, the classic click-and-drag game, will be used in this application. Ideally, in interest of ease of use and simplicity, DAP will be able to run in a similar fashion to a Desktop application, with a clickable file that launches the game, ready for the player to dive in to the exciting and tantalizing world of the Disgruntled Avians project. Finally, and most importantly, our development team at The RAMones are excited for all people ages 4-65[1] to *Dare to Dap*™ with us.

[1] Target Demographic

User Acceptance Testing Matrix

The test matrix enumerates tests to be conducted to verify that the delivered product meets the requirements as stated by the BRD. Tests should be done without the need for understanding the underlying methods of the program.

Feature Matrix

ID	Test Name	Comment	BRD ID
1	Windows Startup		s.1, s.2
2	Apple Startup		s.1, s.2
3	Linux Startup		s.1, s.2
4	Level Select		e.2, e.3, e.12, e.9, e.10, e.12, e.4,
5	Physics Engine		e.5, e.6, e.7,
6	Interfaces		ux.1, ux.2, e.8, e.13
7	Saves		e.1, e.6, e.14,
8	Error Handling		ux.3
9	Obfuscation		s.1

Test Steps

1 - Windows Startup

Process

1. Install a DAP game using the process in the user documentation on a Windows computer
2. Start the game using the process in the user documentation
3. Verify the Start Menu is displayed
4. Verify menu for selecting saves is displayed
5. Select save menu option
6. Verify that the game displays
7. Click and drag avian

8. Verify the GUI updates Avian location
9. Select Menu Button
10. Select each button, verify pause, resume, save, load, help menu function
11. Select quit option

Success:

- Game opens without errors
- Opening the game is accomplished without manual step to select level
- StartMenu, SaveMenu, GameGUI displays
- Pause, resume, save, load, help menu function correctly
- Object movement displays
- Avian launches correctly
- Game exits without error

2 - Apple Startup

Process

1. Install a DAP game using the process in the user documentation on a Windows computer
2. Start the game using the process in the user documentation
3. Verify the Start Menu is displayed
4. Verify menu for selecting saves is displayed
5. Select save menu option
6. Verify that the game displays
7. Click and drag avian
8. Verify the GUI updates Avian location
9. Select Menu Button
10. Select each button, verify pause, resume, save, load, help menu function
11. Select quit option

Success:

- Game opens without errors
- Opening the game is accomplished without manual step to select level
- StartMenu, SaveMenu, GameGUI displays
- Pause, resume, save, load, help menu function correctly
- Object movement displays
- Avian launches correctly
- Game exits without error

3 - Linux Startup

Process

1. Install a DAP game using the process in the user documentation on a Windows computer
2. Start the game using the process in the user documentation
3. Verify the Start Menu is displayed
4. Verify menu for selecting saves is displayed
5. Select save menu option
6. Verify that the game displays
7. Click and drag avian
8. Verify the GUI updates Avian location
9. Select Menu Button
10. Select each button, verify pause, resume, save, load, help menu function
11. Select quit option

Success:

- Game opens without errors
- Opening the game is accomplished without manual step to select level
- StartMenu, SaveMenu, GameGUI displays
- Pause, resume, save, load, help menu function correctly
- Object movement displays
- Avian launches correctly
- Game exits without error

4 - Level Select

Process

1. Using the process in the user documentation, select a level
2. Verify all buttons in Menu option function correctly
3. Verify Avian launches correctly
4. Quit game

Success:

- Game opens without errors
- StartMenu, SaveMenu, GameGUI displays
- Pause, resume, save, load, help menu function correctly
- Object displays correctly
- Object positioned correctly

- Avian launches correctly
- Game exits without error

5 - Physics Engine

Process

1. Start the game using the process in user documentation
2. Select level
3. Click and drag the avian, release to launch
4. Verify Avian Launches
5. Verify Avian collide with object
6. Verify object collide
7. Verify all objects have exit velocity
8. Verify gravity is present
9. Exit game

Success:

- Game runs without errors
- Avian launches correctly
- Avian has gravity
- Avian collides with object
- Objects collide with subsequent objects
- Objects have exit velocity
- Avian has exit velocity

6 - Interfaces

Process

1. Start the game using the process in user documentation
2. Verify the game has started
3. Select both save and load prompt using process in user documentation
4. Verify both buttons function correctly
5. Select a level using process in user documentation
6. Verify all levels display correctly
7. Click and drag avian, release as described in user documentation
8. Verify all objects position in GUI change correctly
9. Open the menu button
10. Select all buttons within the menu button
11. Verify all buttons function as intended
12. Exit game

Success:

- Game opens without errors
- StartMenu, SaveMenu, GameGUI displays
- Pause, resume, save, load, help menu display correctly
- Pause, resume, save, load, help menu function correctly
- Object displays correctly
- Object positioned correctly
- Avian launches correctly
- Game exits without error

7 - Saves

Process

1. Start the game using the process in the user documentation
2. Select “New Player” in the Start Menu
3. Enter name
4. Exit game
5. Start the game using the process in the user documentation
6. Select “Load player” in the Start Menu
7. Enter name used for save player
8. Verify game is loaded

Success:

- Game opens without errors
- On load, player is on same level as previous
- Game exits without error

8 - Error Handling

Process

1. Install Crasher Map
2. Select Crasher Map
3. Observe human readable error message
4. Select technical error report
5. Observe technician readable error report is produced
6. Check that program is no longer running

Success:

- Game produces non-technical message
- Game produces a more detailed technical error report
- Game exits after the error message and report are produced

9 - Obfuscation

Process

1. Run the game and create a save using the user documentation
2. Open game bundle and open map folder with levels
3. Verify each level is not plaintext
4. Verify each save is no plaintext

Success:

- Level files not human readable
- Save files not human readable