RAMones Co.

Disgruntled Avians Project

(DAP)

Prepared by:

Stuart Sessions

Bryan Moreno

Dan Phuong

Ai Ke Woods

Product Requirements Document

Release Date September 6, 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 expanding into the mobile market, micro-transactions, player vs. player game modes, and updates focusing specifically on the physics modeling of DAP.

This document provides nontechnical information regarding the software design of DAP.

Document Versioning

Date	Owner	Comment
09/04/22	Stuart Sessions	Document creation and initial goals.
09/05/22	Ai Ke Woods	Initial Feature Matrix & Discussion
09/06/22	Stuart Sessions	Feature Matrix & Discussion, Document Formatting

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 interface, the classic click-and-drag mode, shall 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 RAMones are excited for all people ages 4-65[1] to *Dare to Dap*™ with us.

[1] Target Demographic

Features

The feature matrix enumerates the features requested for the project and the discussion section provides details regarding the intent of the feature. The ids will be used for traceability. Features that all stakeholders have agreed can be removed should strike-through the feature id and have a comment added to discuss the feature being dropped.

Priority Codes:

H - High, a must have feature for the product to be viable and must be present for launch

M - Medium, a strongly desirable feature but product could launch without

L - Low, a feature that could be dropped if needed

Feature Matrix

ID	Pri	Feature Name	Owner	Comment
S.1	H	Multiplatform	Sales	
S.2	M	Stand-alone application	Sales	
ux.1	H	GUI support	Design	
ux.2	H	User-GUI interaction	Design	
e.1	H	Save/Load game	Creative	
e.2	H	Pausable game loop	Creative	
e.3	H	2D map	Creative	
e.4	M	Progressive levels	Sales	
e.5	H	Physics engine	Creative	
ux.3	H	GUI error notification	Design	
e.6	M	Environment destruction	Creative	
e.7	H	Object collisions	Creative	
e.8	H	Menu	Creative	

e.9	H	Avians	Creative	
e.10	M	Avian trajectory indicator	Creative	
e.11	L	Launch cancel	Creative	
e.12	H	Geometry objects	Creative	
e.13	L	Sprite selection	Creative	
e.14	H	Win condition	Creative	

Feature Discussion

S.1 - Multiplatform

DAP should be downloadable on multiple different platforms to maximize the marketable audience.

S.2 - Stand-alone Application

In order to make it easier for new users to download and play, the app should be able to operate without the need for outside plugins.

U.X.1 - GUI Support

In order to be more compatible with our target audience and more visibly appealing, utilizing a GUI is essential to make the game more entertaining and marketable. The GUI will be the only way this program can be run.

U.X.2 - User-GUI Interaction

When building a game with a GUI, the player must be able to interact with the GUI as part of the game loop. The user should be able to save the game and input actions through the GUI.

E.1 - Save/Load Game

Multiple players may want to play this game on the same device, so it is important that DAP has the ability to save and load different games.

E.2 - Pausable Game Loop

This should Generate the Avians within the game. It will initially be one Avian but if time is available more can be implemented. At some point in development, the game may be able to autosave at certain checkpoints in-between levels.

E.3 - 2D Map

The game will run on a 2D map that can contain different terrain objects as well as an Avian.

E.4 - Progressive Levels

Having multiple levels will give players a goal to work towards in the game, as well as variability. We believe that implementing a progression system by providing different maps would help enrich a user's gaming experience and retain player interest.

E.5 - Physics Engine

The backbone of a destruction game like DAP is the way objects interact with each other, which requires DAP to have a solid physics engine that can be used by the Avian and Objects. Augmenting the sophistication and complexity of the in-game physics engine as the game development progresses will allow the game to have more diverse levels, and better Object collisions.

U.X.3 - GUI Error Notification

For errors that occur when utilizing GUI implementations, a pop up error message will appear for the user that is comprehensible and manageable within the program.

E.6 - Environment Destruction

In addition to disrupting the location of Objects during a game, players will have the opportunity to channel their destructive tendencies in a healthy manner by destroying Objects in the game. Objects that the player is tasked with hitting with their Avian will have a system of hitpoints, and once those hit points are depleted the Object will be destroyed.

E.7 - Object Collisions

For DAP to have an engaging game experience, DAP's physics engine will model the collision between the Avian and the Objects consistently, implementing a simplified version of physics that will represent DAP's unique and constantly evolving game logic.

E.8 - Menu

A list of options describing what the player can do will be presented at the start of the game session. These options will offer ways that the player can navigate through different levels, and in future updates the menu will be where new game modes and features can be selected.

E.9 - Avian

The Avian in DAP will be the player's main method of interacting and progressing through the DAP game levels. Launching the Avian into Objects will clear the levels depending on the win condition setup.

E.10 - Avian Trajectory Indicator

Game should show where the Avian will go, displaying launch information such as angle.

E.11 - Launch Cancel

If the player decides to not launch the Avian, they can cancel their action.

E.12 - Geometry Objects

The game should generate Objects that can react to Avian collisions and are updated accordingly by Physics Engine.

E.13 - Sprite Selection

The player should be able to select from a set of given Sprites (images to be associated with the player's Avian) and use that sprite to interact with game and game levels.

E.14 - Win Condition

A win condition for each game level is required to flag a level as cleared. The win condition should be an Avian coming into contact with target Object or target Object touching the floor. A win condition for the game would be to clear every level given. Future win conditions could be subject to change.

User Stories

User stories should never be removed from the document. If stakeholders agree to remove a user story, the user story name should be changed to strike-through and a comment regarding the approval to drop the user story must be provided. The primary users of DAP are game players. As to who can pick up a game, that could be anyone, and we can envision a variety of different people playing DAP. People who grew up with a tablet since they were a kid, competitive gamers looking to make a name for themselves, or even people who still pronounce Google as goggle. RAMones Co. strives to make the joy of DAP's gameplay and microtransactions available to all people.

Mirabel, a Cartoon Lover.
Mirabel is 5 years old. She likes playing games and especially loves cute cartoon characters. She likes simple puzzle games and wants to be able to succeed at games by herself. Mirabel likes playing games but she gets frustrated if she can't understand how to win.

Bartholomew, an Influencer
Barth is a 15 year old student. He wants the game to provide ridiculous images with memes or other entertaining images colliding within the game. He wants the game to be funny so he can post it on his YouTube channel.

Carly, a Mom
Carly is a 31 year old mother to a 4 year old child. She wants a child-appropriate game that her child can play during his "game time" without her worrying about what's on it.

Jeff, a Grandpa
Jeff is a 65 year old Grandfather. His toddler grandson recently started to like playing video games. Jeff wants a simple game that he can learn and help his Grandson to play.

Use Case 1

Name	Start new game
ID	UC01
Description	User runs the program for the first time
Actor	Player
Organizational	User needs to be able to run the game and save/load their file

Benefits	
Use Frequency	Very Frequent
Triggers	User loads application
Preconditions	GUI Displays correctly, shows load options
Main success scenarios	● User selects new player ● User enters player name/id ● System saves player name
Extensions (Error Scenarios)	● User enters blank name ● System fails to load/save ● System redirects user to enter player name OR ● New user selects returning player ● New user enters non-existent player name ● System fails to load ● System redirects user to enter player name
Alternative Courses	● User selects returning player ● User enters play name/id ● System loads player data
Post conditions	Application loads prompt for level choice

Use Case 2

Name	Selecting level choice
ID	UC02
Description	User selects level
Actor	Player
Organizational Benefits	User needs to be able to run the game, choose level, and run correctly
Use Frequency	Very Frequent
Triggers	User finishes loading/saving file
Preconditions	GUI Displays correctly, player has loaded/started new save file
Main success	● User selects level choice

scenarios	● Application loads level choice ● Level choice is correct
Extensions (Error Scenarios)	● User selects level choice ● Application fails to load correct level choice ● Application redirects user to pick again
Alternative Courses	● User selects exit program ● Program exits
Post conditions	Level is presented on GUI interface

Use Case 3

Name	Launching the Avian
ID	UC03
Description	User aims and fires the Avian
Actor	Player
Organizational Benefits	User needs to be able to launch the Avian to play the game
Use Frequency	Very Frequent
Triggers	Level loads/User misses
Preconditions	GUI Displays correctly,
Main success scenarios	● User clicks on their Avian ● User drags the Avian backwards ● User releases the Avian and it is launched at a speed and angle relative to how the avian was pulled back
Extensions (Error Scenarios)	● User immediately releases the Avian ● User drags the Avian Forwards and then releases
Alternative Courses	● System Prompts user to try again explaining that the Avian must be pulled back and then released.
Post conditions	Physics Engine is loaded

Use Case 4

Name	Physics Engine handles correctly
ID	UC04
Description	Physics Engine must handle interactions with objects correctly
Actor	Player, Application Objects
Organizational Benefits	Physics engine must handle interactions correctly in order for the game to run as intended.
Use Frequency	Very Frequent
Triggers	User launches Avian
Preconditions	Level choice loaded correctly, Avian launched
Main success scenarios	● User launches Avian ● Avian travels in correct trajectory ● Avian collides with Objects, dispersing
Extensions (Error Scenarios)	● User launches Avian ● Avian does not follow expected trajectory ● Application redirects user to launch again OR ● User launches Avian ● Avian does not launch ● Application redirects user to launch again OR ● User launches Avian ● Avian follows correct trajectory ● Avian collision with objects not handled correctly OR ● User launches Avian ● Avian follows correct trajectory ● Application collisions with objects handled correctly ● Objects collided with do not disperse correctly
Alternative Courses	● User launches Avian ● Avian follows correct trajectory ● User misses ● Application prompts to launch again

Post conditions	Objects disperse or player misses

Use Case 5

Name	Cleared Level
ID	UC05
Description	Win condition for clearing a level must be handled correctly
Actor	Player
Organizational Benefits	Correct win condition allows user to progress through levels, encouraging interaction with application and perhaps sales
Use Frequency	Somewhat Frequent
Triggers	When all objects are stationary for more than 3 seconds, 15 seconds have passed after launching, or run out of Avians
Preconditions	All objects stationary for 3 seconds, 15 seconds have passed, or no more Avians to launch, or Objective filled
Main success scenarios	● Preconditions filled ● Clear level check runs ○ Application redirects to win screen if Objectives filled ○ Application redirects to lose screen if no more Avians to launch and objective isn't filled
Extensions (Error Scenarios)	● Preconditions filled ● Clear level check does not run OR ● Clear level check does not run ● Application redirects to win screen if Objectives aren't filled ● Application redirects to lose screen if Objectives filled ● Applications redirects to launch again if no more Avians present
Alternative Courses	● Preconditions filled ● Clear level check runs ● Application redirects to launch again if player still has Avians to launch
Post conditions	Prompts to win screen or lose screen, loads Level Complete Option screen

Use Case 6

Name	Level Complete Option Screen
ID	UC06
Description	User decides how to proceed after clearing the level
Actor	Player
Organizational Benefits	User needs to be able to select how to proceed after clearing a level
Use Frequency	Somewhat Frequent
Triggers	Level is Cleared
Preconditions	Level is Cleared
Main success scenarios	● User is presented with a GUI offering Retry Level, Quit Game, Select New Level, and Save Game ● User selects one of the options ● Redirects user based on their selection ● Level Restarts
Extensions (Error Scenarios)	● Application redirects to the wrong screen ● Application fails to redirect ● Application fails to Quit Game
Alternative Courses	● Save Game ● Level is not cleared
Post conditions	Application loads prompt for level choice

Use Case 7

Name	Win Game
ID	UC07
Description	Check to see if all levels have been cleared
Actor	Player

Organizational Benefits	Incentives player to clear all levels
Use Frequency	Not Frequent
Triggers	Check WinGame is used
Preconditions	All levels must be cleared
Main success scenarios	● All levels cleared ● Win Game prompt appears
Extensions (Error Scenarios)	N/A
Alternative Courses	N/A
Post conditions	User can replay levels if they wish

Use Case 8

Name	QuitGame
ID	UC08
Description	Application quits game
Actor	Player
Organizational Benefits	Allows player to quit game
Use Frequency	Frequent
Triggers	Exit button
Preconditions	While application is running
Main success scenarios	● User selects to quit the game ● User is prompted to save ● System saves ● Application quits
Extensions (Error	● User selects to quit the game ● User is prompted to save

Scenarios)	● System does not save ● Application quits
Alternative Courses	● User selects to quit the game ● User is prompted to save ● User chooses to not save
Post conditions	System is saved if user elects to save

Use Case 9

Name	Help
ID	UC09
Description	Help menu for what to do
Actor	Play
Organizational Benefits	Lets user know how to play the game
Use Frequency	Low frequency
Triggers	Help button
Preconditions	Application loaded
Main success scenarios	● User selects help ● Help Menu displays
Extensions (Error Scenarios)	N/A
Alternative Courses	User knows how to play game completely unassisted
Post conditions	Help display demonstrates how to play the game

Use Case 10

Name	Save Game

ID	UC10
Description	User saves progress in application
Actor	Player
Organizational Benefits	Incentives the player to continue playing the game if they can save
Use Frequency	Very frequent
Triggers	User presses button to save
Preconditions	User cannot be in the middle of launching an Avian
Main success scenarios	● User presses save button ● User enters in player name ● System saves file
Extensions (Error Scenarios)	● User presses save button ● User enters blank name ● System fails to save ● System redirects to press save again
Alternative Courses	N/A
Post conditions	Game is now saved onto computer

Use Case 11

Name	Load game
ID	UC11
Description	User wants to load in their file
Actor	Player
Organizational Benefits	Encourages player to continue playing to continue their progress
Use Frequency	Frequent
Triggers	User presses load file button
Preconditions	Application loaded

Main success scenarios	● User presses load game ● User enters name ● Game file is restored and loaded onto application
Extensions (Error Scenarios)	● User presses load game ● User enters invalid name ● Game file not loaded ● Application prompts user to select name
Alternative Courses	N/A
Post conditions	Previous save is now loaded