Test Report

Gill, Surinder
1308896

Hu, Joshua
1311940

Lago, Nick
1302613

November 27, 2015

Contents

List of Figures

List of Tables

1 Revision History

Table 1: Revision History: Proof of Concept Plan

DATE	DEVELOPER	CHANGE	REVISION
November 25, 2015	Gill, Surinder	Initial Draft	0
November 25, 2015	Hu, Joshua	Initial Draft	0
November 25, 2015	Lago, Nick	Initial Draft	0

2 General Information

2.1 Summary

This document is intended to provide a complete encapsulation of the results of the testing that was performed during the development of and indicated in the Test Plan of the FloppyFish project.

Specifically the document entails the details of the tests performed on specific sections of the code. The code functionality which was tested includes but is not limited to the business functions of the project, i.e. game logic and data interaction, rendering of the game objects, and support functionality.

2.2 Environment and Pretest Background

Floppy Fish is a project who's aim is to redevelop an open source implementation of Flappy Bird with proper documentation of project development principles. It was developed over the past few months and has completed its initial testing phase with future rounds of development and testing to come.

The testing has been conducted by the development team (the A Team) and has been conducted on the team's local machines.

Testing during development occurred, usually under manual structural testing in order to develop a working implementation, but no prior testing that would affect the official testing phase occurred.

2.3 Test Objectives

The testing phase's motivation is to uncover and confirm functional and non-functional requirements of the project.

Specifically the testing is to ensure the validity of the implementation and its correctness (adherence to the requirements).
This includes:

- Playable game mechanics
- Expected rendering of game media
- Expected game behaviour

2.4 Expected Defect Rates

The number of expected defects found by testing is to be or less than 10% of the lines of project code. Those defects are expected to be the source of errors rather than an assumption of their propagation through the rest of the expected behaviour.

2.5 References

Not applicable.

3 Plan

3.1 Software Description

Table 2: Function Overview

Item No.	Function	Input	Output
1	Draw.rect	Integers	
2	Draw.circle	Integers	
3	Draw.Image	image + Integers	
4	Draw.Sprite	image + Integers	
5	Draw.text	String + Integers	
6	Input.set	mouse click	
7	requestAnimFrame		function
8	BottomBar.update		
9	BottomBar.respawn		
10	BottomBar.render		
11	Pipe.update		
12	Pipe.render		
13	Pipe.respawn		
14	Pipe.randomIntFromInterval	Integers	Integer
15	fish.update		
16	fish.powerMode		
17	fish.render		
18	PowerUp.update		
19	PowerUp.render		
20	PowerUp.respawn		
21	PowerUp.randomIntFromInterval	Integers	Integer
22	PowerUp.Remove		
23	Particle.Update		
24	Particle.render		
25	Collides	fish + pipe	boolean
26	collidesPowerUp	fish +powerup	boolean
27	Splash.init		
28	Splash.update		
29	Splash.render		
30	Play.init		
31	Play.update		
32	Play.render		
33	GameOver.getHighScore		Integer
34	GameOver.init		
35	GameOver.updated0		
36	GameOver.render		

Descriptions of each function:

1. Draws a rectangle at coordinates input
2. Draws a circle at coordinates input
3. Draws a image at coordinates input, from the given image
4. Draws a sprite using some inputs as where to cut the image and others as where to draw it. The image is taken in as an input
5. Takes in a string and draws that string at the coordinates given
6. Checks to see if the user has clicked within the screen
7. Initializes the bar on the bottom of the screen.
8. requests the animation frame for sizing purposes.
9. Updates x coordinate of the bottom bar
10. sets the bottom bar at a new x location (so it doesn‘t run off the screen)
11. uses draw to render the bottom bar
12. Updates Pipes variables (checking if the pipe needs to respawn)
13. uses draw to render the pipe
14. resets the Pipes coordinate variables to the beginning of the screen and sets the coin as true
15. picks a random integer to be used as the location of the gap (so it‘s always changing)
16. this will act similar to respawn except it will move the powerup icon off the screen
17. this will update the power ups x coordinate and check if it needs to respawn
18. uses draw functions to create the power up icon
19. will update the coordinates of the power up so it is on the screen
20. will return a random integer within the interval
21. will update particles position
22. creates the particle
23. checks for collisions between a fish and a pipe

24. checks for collision between a powerup and a fish
25. initializes the splash mode of playing the game (awaiting user input)
26. update will wait for the user to click their mouse
27. render will call all of the entities (where applicable) and access their rendering functions
28. initializes the play state of the game
29. accesses the update funciton of all entities that have been pushed
30. accesses the rendering funcitons of all pushed entities
31. will access the high score saved by cookies
32. will access all of the update functions by all pushed entities (like bottom bar)
33. will access all render functions by all pushed entities

3.2 Test Team

Table 3: Test Team

DEVELOPER	**ROLE**
Gill, Surinder	Functional Testing
Hu, Joshua	Structural Testing
Lago, Nick	Functional Testing

3.3 Milestones

Table 4: Testing Milestones

Event No.	Event	Start Date	End Date
1	Initial Development Testing	11/05/15	11/05/15
2	Survey Round 1 Implementation Testing	11/10/15	11/15/15
3	Survey Round 2 Implementation Testing	11/17/15	11/29/15

3.4 Budgets

Not applicable.

3.5 Initial Testing (Systems Checkpoint)

3.5.1 Schedule

Table 5: Testing Milestones

Event No.	Event	Start Date	End Date	Resources
1	Training	11/01/15	11/03/15	JavaScript Testing Tools
2	Test Design	11/04/15	11/04/15	
3	Testing	11/17/15	11/29/15	JavaScript Testing Tools
1	3.3.1 Testing	11/05/15	11/05/15	JavaScript Testing Tools
2	3.3.2	11/10/15	11/15/15	JavaScript Testing Tools
3	3.3.3	11/17/15	11/29/15	JavaScript Testing Tools

3.5.2 Requirements

(a) Equipment

Team members machines will be required for the duration of testing, and each member may have any number and type of machine, with a minimum of one machine.

(b) Software

- Test-Driver
- Jasmine
- JSDocs

(c) Personnel

Gill, Surinder:

- Java Script Developer
- Available for testing

Hu, Joshua:

- Java Script Developer
- Available for testing

Lago, Nick:

- Java Script Developer
- Available for testing

3.5.3 Testing Materials

(a) System Documentation

- JSDocs

(b) Software-To-Be Tested and Its Medium

- JavaScript
- In HTML file loaded by browser

(c) Test Inputs

- Tester designed data

(d) Test Documentation

- JS TEST DOCS

(e) Test Tools

- Test-Driver, Jasmine

3.5.4 Test Training

The personnel to be trained are the developer and test teams, who will be trained to test JavaScript in console checks, Unit Testing through JavaScript testing frameworks such as PALCEHOLDER. The training will be highly self motivated and executed.

3.5.5 Test-To-Be Conducted

The tests conducted at this point will be comprehensive in the scope of the project, covering both structural and unit tests through manual and automated means.

3.6 Continuing Testing (Systems Checkpoint)

For all subsequent testing, extensive testing will be extensive on the implemented changes, and unchanged areas of the project will be tested once and documented as a pass or not to confirm they are still valid.

4 Specifications and Evaluation

4.1 Specifications

4.1.1 Business Functions

- The executable HTML file will create a new browser window.

Fit Criterion or Test Case:
Is a new browser window created upon the execution of the HTML file?

- The HTML will be executed by a browser with JavaScript functionality.

  Fit Criterion or Test Case:
  Attempt to execute the HTML file with multiple major browsers with HTML functionality.

- The game will have a standby state in which it waits for user input.

  Fit Criterion or Test Case:
  Execute game and given an arbitrary timeframe if the game does not produces any unexpected action during that timeframe the game does not respond without user input.

- Upon the reception of user input from the standby state the game will begin.

  Fit Criterion or Test Case:
  Provide user input to check if the state changes.

- At the beginning of the game the user will perceive all stats reset to their default state.

  Fit Criterion or Test Case:
  Return the values of all stats upon the change from the default state.

- At the beginning of the game the user character will maintain its state until user input is received.

  Fit Criterion or Test Case:
  Return relative character position at a regular interval prior to and during user input.

- If there is a collision with the user character and an obstacle object the game will terminate and all stats will be recorded.

  Fit Criterion or Test Case:
  Given an arbitrary True value of a collision check see if the state changes and return the values of the stats.

- Upon termination of the game state all stats will be reset to their default state and the standby state will be reinitiated.

  Fit Criterion or Test Case:
  Return the values of the stats and check the state.

- If there is a collision with the user character and an objective object the user's score will increment and the objective object's instance will terminate.

Fit Criterion or Test Case:
Return the user's score, check the object's instance.

- During the game state reception of user input will cause the user character to respond in a constant and uniform manner relative to the user character's instance.

  Fit Criterion or Test Case:
  Check the response of the user character.

4.1.2 Structural Functions

1. Flappy_Fish.Draw.rect
2. Flappy_Fish.Draw.circle
3. Flappy_Fish.Draw.Image
4. Flappy_Fish.Draw.Sprite
5. Flappy_Fish.Draw.text
6. Flappy_Fish.BottomBar.update
7. Flappy_Fish.BottomBar.render
8. Flappy_Fish.BottomBar.respawn
9. Flappy_Fish.Pipe.update
10. Flappy_Fish.Pipe.render
11. Flappy_Fish.Pipe.respawn
12. Flappy_Fish.fish.update
13. Flappy_Fish.fish.render
14. Flappy_Fish.fish.powerMode
15. Flappy_Fish.PowerUp.update
16. Flappy_Fish.PowerUp.render
17. Flappy_Fish.PowerUp.respawn
18. Flappy_Fish.PowerUp.Remove
19. Flappy_Fish.Collides

20. Flappy_Fish.CollidesPowerUp
21. Flappy_Fish.Splash.init
22. Flappy_Fish.Splash.update
23. Flappy_Fish.Splash.render
24. Flappy_Fish.Play.init
25. Flappy_Fish.Play.this.update
26. Flappy_Fish.Play.render
27. Flappy_Fish.GameOver.getMedal
28. Flappy_Fish.GameOver.getHighScore
29. Flappy_Fish.GameOver.init
30. Flappy_Fish.GameOver.update
31. Flappy_Fish.GameOver.render

4.1.3 Test/Function Relationships

- Manual Game Play & Project Execution
  - 4.1.2.1
  - 4.1.2.2
  - 4.1.2.3
  - 4.1.2.4
  - 4.1.2.5
  - 4.1.2.6
  - 4.1.2.7
  - 4.1.2.8
  - 4.1.2.9
  - 4.1.2.10
  - 4.1.2.11
  - 4.1.2.12
  - 4.1.2.13
  - 4.1.2.14

- 4.1.2.15
- 4.1.2.16
- 4.1.2.17
- 4.1.2.18
- 4.1.2.19
- 4.1.2.20
- 4.1.2.21
- 4.1.2.22
- 4.1.2.23
- 4.1.2.24
- 4.1.2.25
- 4.1.2.26
- 4.1.2.27
- 4.1.2.28
- 4.1.2.29
- 4.1.2.30
- 4.1.2.31

4.1.4 Test Progression

Tests will be conducted in order of function criticality. This means that testing of the core game engine will be conducted as a priority, then secondary functions such as rendering, then auxiliary functions such as audio playing. In these differing levels of criticality testing order will be done on dependencies, that is functions which call other functions will be tested after the testing of the functions called have been tested.

4.2 Methods and Constraints

4.2.1 Methodology

The method of testing for this project is to approach it from testing the core functionality is valid, then test to ensure the non-functional requirements are met to developer satisfaction (meeting of critique and feedback will influence the developers' notion of satisfaction).

4.2.2 Test Tools

Specify the type of the test tools to be used.

- Jasmine
- JavaScript console
- Javascript testing framework

4.2.3 Extent

The testing for this project will be near total, excluding minor syntactical code, as the project is smaller in scope and magnitude, and feasible under our circumstances.

4.2.4 Data Recording

An HTML page will be used to record all test results.

4.2.5 Constraints

Not applicable.

4.3 Evaluation

4.3.1 Criteria

Tests will be conducted using fringe cases and exceptions as tests for the limits of the functionality and will also test using a small set of "normal" cases.

4.3.2 Data Reduction

In the case of unit testing, tests will be represented by boolean values to indicate their passing status. This simplifies the volume of data needed to be considered in a final abstracted view. Structural tests will also be summarized in this manner.

5 Test Descriptions

This link contains the unit tests conducted on the code. http://surinderdemo.github.io/FloppyFishDemo/

5.1 Window Setter Unit Tests

5.1.1 Control

Return an appropriate animation frame

5.1.2 Inputs

Describe the input data and input commands used during the test.

5.1.3 Outputs

Animation Frame

5.1.4 Procedures

Specify the step-by-step procedures to accomplish the test. Include test setup, initialization, steps and termination.

5.2 Cookie Handler Unit Tests

5.2.1 Control

Set Cookie Name, Value and Expiry Days Cookie should not be set at 2 Get Cookie that was previously set at 42 Check if cookies are not Null

5.2.2 Inputs

Describe the input data and input commands used during the test.

5.2.3 Outputs

Describe the output data expected as a result of the test and any intermediate messages that may be produced.

5.2.4 Procedures

Specify the step-by-step procedures to accomplish the test. Include test setup, initialization, steps and termination.

5.3 Main Bottom Bar Unit Tests

5.3.1 Control

Check if background has been uploaded. Check coordinates to draw the background.

5.3.2 Inputs

Describe the input data and input commands used during the test.

5.3.3 Outputs

Describe the output data expected as a result of the test and any intermediate messages that may be produced.

5.3.4 Procedures

Specify the step-by-step procedures to accomplish the test. Include test setup, initialization, steps and termination.

5.4 Main Collides Unit Test

5.4.1 Control

Check to see if collision with pipe, coin, or ground happened Check to see if distances match each other Check to see if collision is null

5.4.2 Inputs

Describe the input data and input commands used during the test.

5.4.3 Outputs

Describe the output data expected as a result of the test and any intermediate messages that may be produced.

5.4.4 Procedures

Specify the step-by-step procedures to accomplish the test. Include test setup, initialization, steps and termination.

5.5 Continuing Tests

Describe the second and subsequent tests in a manner similar to that used in paragraph 4.1.

5.6 Main Collides Power Up Unit Tests

5.6.1 Control

Check to see if collision with pipe, coin, or ground happened Check to see if distances match each other Check to see if collision is null

5.6.2 Inputs

Describe the input data and input commands used during the test.

5.6.3 Outputs

Describe the output data expected as a result of the test and any intermediate messages that may be produced.

5.6.4 Procedures

Specify the step-by-step procedures to accomplish the test. Include test setup, initialization, steps and termination.

5.7 Continuing Tests

Describe the second and subsequent tests in a manner similar to that used in paragraph 4.1.

5.8 Main Draw Unit Tests

5.8.1 Control

Create Rect Function and check if null Create Circle Function and check if null Create Image Function and check if null Create Sprite Function and check if null Create Text Function and check if null

5.8.2 Inputs

Describe the input data and input commands used during the test.

5.8.3 Outputs

Describe the output data expected as a result of the test and any intermediate messages that may be produced.

5.8.4 Procedures

Specify the step-by-step procedures to accomplish the test. Include test setup, initialization, steps and termination.

5.9 Continuing Tests

Describe the second and subsequent tests in a manner similar to that used in paragraph 4.1.

5.10 Main Fish Unit Test

5.10.1 Control

Check if image is loaded Check for Gravity instance Check for velocity instance Check if user has tapped Confirm the jump with jump buffer

5.10.2 Inputs

Describe the input data and input commands used during the test.

5.10.3 Outputs

Describe the output data expected as a result of the test and any intermediate messages that may be produced.

5.10.4 Procedures

Specify the step-by-step procedures to accomplish the test. Include test setup, initialization, steps and termination.

5.11 Continuing Tests

Describe the second and subsequent tests in a manner similar to that used in paragraph 4.1.

5.12 Main Gameover Tests

5.12.1 Control

Check for medal object returned Get the x and y values for the input Check to see if cookie was stored

5.12.2 Inputs

Describe the input data and input commands used during the test.

5.12.3 Outputs

Describe the output data expected as a result of the test and any intermediate messages that may be produced.

5.12.4 Procedures

Specify the step-by-step procedures to accomplish the test. Include test setup, initialization, steps and termination.

5.13 Continuing Tests

Describe the second and subsequent tests in a manner similar to that used in paragraph 4.1.

5.14 Main Input Tests

5.14.1 Control

Check for x and y values on input when tapped Confirm that tap is undefined when not clicked

5.14.2 Inputs

Describe the input data and input commands used during the test.

5.14.3 Outputs

Describe the output data expected as a result of the test and any intermediate messages that may be produced.

5.14.4 Procedures

Specify the step-by-step procedures to accomplish the test. Include test setup, initialization, steps and termination.

5.15 Continuing Tests

Describe the second and subsequent tests in a manner similar to that used in paragraph 4.1.

5.16 Main Pipe Unit Tests

5.16.1 Control

Check for center of the pipe value Get the coin value for the pipe

5.16.2 Inputs

Describe the input data and input commands used during the test.

5.16.3 Outputs

Describe the output data expected as a result of the test and any intermediate messages that may be produced.

5.16.4 Procedures

Specify the step-by-step procedures to accomplish the test. Include test setup, initialization, steps and termination.

5.17 Continuing Tests

Describe the second and subsequent tests in a manner similar to that used in paragraph 4.1.

5.18 Main Play Unit Tests

5.18.1 Control

Describe the test control, such as manual, semiautomatic or automatic insertion of inputs, sequencing of operations, and recording of results.

5.18.2 Inputs

Describe the input data and input commands used during the test.

5.18.3 Outputs

Describe the output data expected as a result of the test and any intermediate messages that may be produced.

5.18.4 Procedures

Specify the step-by-step procedures to accomplish the test. Include test setup, initialization, steps and termination.

5.19 Continuing Tests

Describe the second and subsequent tests in a manner similar to that used in paragraph 4.1.

5.20 Main Power Up Unit Tests

5.20.1 Control

Describe the test control, such as manual, semiautomatic or automatic insertion of inputs, sequencing of operations, and recording of results.

5.20.2 Inputs

Describe the input data and input commands used during the test.

5.20.3 Outputs

Describe the output data expected as a result of the test and any intermediate messages that may be produced.

5.20.4 Procedures

Specify the step-by-step procedures to accomplish the test. Include test setup, initialization, steps and termination.

5.21 Continuing Tests

Describe the second and subsequent tests in a manner similar to that used in paragraph 4.1.

5.22 Main Splash Unit Tests

5.22.1 Control

Describe the test control, such as manual, semiautomatic or automatic insertion of inputs, sequencing of operations, and recording of results.

5.22.2 Inputs

Describe the input data and input commands used during the test.

5.22.3 Outputs

Describe the output data expected as a result of the test and any intermediate messages that may be produced.

5.22.4 Procedures

Specify the step-by-step procedures to accomplish the test. Include test setup, initialization, steps and termination.

5.23 Continuing Tests

Describe the second and subsequent tests in a manner similar to that used in paragraph 4.1.

5.24 Namespace Unit Tests

5.24.1 Control

Describe the test control, such as manual, semiautomatic or automatic insertion of inputs, sequencing of operations, and recording of results.

5.24.2 Inputs

Describe the input data and input commands used during the test.

5.24.3 Outputs

Describe the output data expected as a result of the test and any intermediate messages that may be produced.

5.24.4 Procedures

Specify the step-by-step procedures to accomplish the test. Include test setup, initialization, steps and termination.

5.25 Continuing Tests

Describe the second and subsequent tests in a manner similar to that used in paragraph 4.1.

5.26 Sound Unit Tests

5.26.1 Control

Check to see if Sound plays Get the current time of Sound played

5.26.2 Inputs

Describe the input data and input commands used during the test.

5.26.3 Outputs

Describe the output data expected as a result of the test and any intermediate messages that may be produced.

5.26.4 Procedures

Specify the step-by-step procedures to accomplish the test. Include test setup, initialization, steps and termination.

5.27 Continuing Tests

Describe the second and subsequent tests in a manner similar to that used in paragraph 4.1.