JAVA PROGRAMMING

EXECUTIVE SUMMARY FOR UNIT TESTING ON THE EGGHUNT PROGRAM

Friday 10th May, 2019

Unit Testing Team:

Cinthya Grajeda

Dhwani Snehalbhai Sheth

Chinthi Reddy Avinash Reddy

Krunal Ghanshyam Mistry

Ashish Shrestha

Vidhur Savyasachin

Instructor:

CTO Wes Kirschner

Battered Egg Incorporated

Executive Summary

As part of the software developing life cycle, the Unit Testing team, performed software unit testing on all five classes and ten main functions encountered in the EggHunt game. Testing of every major unit involved, planning, developing, and performing the tests cases, both, manually and using the JUnit tool. A summary of the major test cases per class is depicted in Table 1.

Class	Test Case/Functions	Expected Result	Actual Result
EggHunt	buildGui() onChanged() doClick()	A GUI A win, lose, or continue scenario A Snake thread A Chick thread	All Passed
Chick	run() layOnGrass()	Add eggs to list, call layOnGrass(), sleep Add eggs to the grass pane	All Passed
Egg	Egg()	Eggs are created with random colors	All Passed
Snake	run() eatEggs() eatFromGrass()	Call eatEggs(), hibernate, die Eat eggs close enough to point p Remove eggs from grass pane	All Passed
MyPoint	distanceSq()	Create point, return distance	All Passed

Table 1: Overall results of major test cases

Results indicate this program works within its limits. As noted, all unit tests were passed. However, when additional tests were conducted, irregularities were discovered, such as maximum and minimum boundaries on parameters. For instance, increasing or decreasing certain values will either help the user or computer win the game easily.

Furthermore, when integrated testing took place, six major bugs were discovered. For example, if the computer wins the game, eggs are still shown on the GUI, the user can still click on those eggs and win the game. Moreover, if the user wins the game, and no eggs are left on the pane, the user can still click on the pane and spawn new snakes. Other bugs include, eggs laying on top or near each other, and appearing at the same time interval on the window. Additionally, if the window's size is maximized, the pane does not bind to it and has no effect on the game.

Consequently, the team developed and implemented unit test cases that exposed the advantages and weaknesses of this game. Major flaws were found that could negatively affect how customers accept and enjoy this game. It is recommended to further analyze the results and use them as a guide that could possibly benefit Battered Egg Inc.'s future software development.