January 26, 2016 | Group 14 members:
EECE514 Assignment 1 | - Babatunde Akinniyi - Cyrus Sassani

Exercise 1.2:
Test Charter: Analyze and test the gameplay of the JPacman

Actor: Novice gamer

Purpose: To explore the JPacman java program and identify potential missing features/functionalities.

Setup: A JPacman Java project imported into Eclipse compiled by Maven. JPacman is run as a Java application in eclipse and then explored. The system running the application is an Apple MacBook Air laptop.

Priority: High. The game is useless if it cannot be played fairly and doesn't behave reasonably.

Reference: https://s3.amazonaws.com/piazza-resources/ij60by7lpxh4n2/ijpz5au1n8o4eo/assignment1.pdf?AWSAccessKeyId=AKIAIEDNRLJ4AZKBW6HA&Expires=1453881867&Signature=F%2FxVkH3AB9WRcU1PKW3DtYIQAmI%3D

Data: Variety of user clicks, drags and maneuvering of the Mr. Pac-Man.

Activities/Test Scenarios:

1. Moving pacman (Landmark tour):
    a. Run the JPacman application
    b. Click start button
    c. Try to move Pac-Man
    d. Result/bug: Pac-Man does not move. Expected behavior: Pac-Man moves.
    e. Click exit button
2. Watching gameplay (Couch potato tour):
    a. Run the JPacman application
    b. Click start button
    c. Watch game play. Study how red players move around maze
    d. Eventually, red player finds Pac-Man and eats him.
    e. Bug: There is no "reset/restart" button. Player always must exit game and re-rerun program.
3. Modify status text field (Antisocial tour):
    a. Run the JPacman application
    b. Click start button
    c. Try to edit status text field
    d. Notice it is un-editable
    e. Result/bug: text field has no descriptive label to user. Expected result: have label for field followed by plain text without textbox field so that user doesn't attempt to edit it.
    f. Bug: When text field is attempted to be edited, the Pac-Man is no longer controllable
    g. Click exit button

Exercise 1.3:

- There are 10 test classes
- There are 39 test cases
- I modified the testInitialSetting() method in the GameTest class. The expected initial direction of the player was changed from LEFT to RIGHT. This caused the JUnit assertEquals() test to fail. All together, there were 12 runs, 1 failure and no error.

Exercise 1.4:

I picked the ButtonPanelTest test class for examination. After examination, it seems very barebones and incomplete. The main purpose of this test class is to test the functionality of the buttons on the Jpacman application frame. The buttonPanelSmokeTest() test method creates a JFrame and a button linked to the parent frame. Next, it adds the button to the frame and makes it visible. Further setup up needs to be done in order to listen in on user clicks on the button.

Some test scenarios that can be run when setup is fully done are:

1. Given that START button is enabled, test that it is clickable and triggers start game action when clicked on.
2. Given that START button is disabled, test that START button is un-clickable and triggers no action or function when button is clicked on.
3. Given that STOP button is enabled, test that it is clickable and triggers stop game action when clicked on.
4. Given that STOP button is disabled, test that STOP button is un-clickable and triggers no action or function when button is clicked on.
5. Given that EXIT button is enabled, test that it is clickable and triggers program termination when clicked on.
6. Given that EXIT button is disabled, test that it is un-clickable and does not trigger program termination when clicked on.