SE 3XA3: Test Report
PineSweeper

Team 7, PineApple
Prince Sandhu (sandhps2)
Vishesh Gulatee (gulatev)
Varun Rathore (rathorvs)

December 8, 2016

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	**Version**	**Notes**
2016-12-01	1.0	Completion of Requirements Evaluations
2016-12-07	1.1	Completion of Unit Testing Section

1 Functional Requirements Evaluation

1.1 Graphical User Interface Testing

1. FR-GUI-Test-1

   Type: Manual, Functional, Dynamic

   Initial State: The main menu or when the game is in progress (timer is running).

   Input: The beginner difficulty level is selected.

   Expected Output: New game with beginner difficulty and corresponding dimensions BDIMENSIONS and corresponding number of mines BMINES.

   Output: New game with beginner difficulty and corresponding dimensions BDIMENSIONS and corresponding number of mines BMINES.

   Result: PASS

2. FR-GUI-Test-2

   Type: Manual, Functional, Dynamic

   Initial State: The main menu or when the game is in progress (timer is running).

   Input: The intermediate difficulty level is selected.

   Expected Output: New game with intermediate difficulty and corresponding dimensions IDIMENSIONS and corresponding number of mines IMINES.

   Output: New game with intermediate difficulty and corresponding dimensions IDIMENSIONS and corresponding number of mines IMINES.

   Result: PASS

3. FR-GUI-Test-3

   Type: Manual, Functional, Dynamic

   Initial State: The main menu or when the game is in progress (timer is running).

   Input: The advanced difficulty level is selected.

   Expected Output: New game with advanced difficulty and corresponding dimensions ADIMENSIONS and corresponding number of mines AMINES.

   Output: New game with advanced difficulty and corresponding dimensions ADIMENSIONS and corresponding number of mines AMINES.

   Result: PASS

4. FR-GUI-Test-4

   Type: Manual, Functional, Dynamic

   Initial State: The game is in progress (timer is running) or the game has terminated (a hidden mine has been clicked upon). Input: New game/reset icon is selected. Expected Output: New game is started, with the same difficulty level as the game that is presently running. The mines shall be in different locations than the previous.

   Output: New game is started, with the same difficulty level as the game that is presently running. The mines shall be in different locations than the previous.

   Result: PASS

5. FR-GUI-Test-5

   Type: Manual, Functional, Dynamic

   Initial State: The game is in progress (timer is running).

   Input: Tile/Button (which contains either: a mine, a number, or a blank) is clicked upon in the PineSweeper Grid.

   Expected Output: Tile/Button disappears and reveals one of the three aforementioned elements hidden beneath it.

   Output: Tile/Button disappears and reveals one of the three aforementioned elements hidden beneath it.

   Result: PASS

6. FR-GUI-Test-6

   Type: Manual, Functional, Dynamic

   Initial State: The game is in progress (timer is running).

   Input: Hidden blank cell is clicked upon.

   Expected Output: All eight bordering cells are displayed and the same process follows, recursively, for each of the neighbouring blank cells.

   Output: All eight bordering cells are displayed and the same process follows, recursively, for each of the neighbouring blank cells.

   Result: PASS

7. FR-GUI-Test-7

   Type: Manual, Functional, Dynamic

   Initial State: The game is in progress (timer is running).

   Input: User clicks on the pause button.

   Expected Output: The in-game timer pauses.

   Output: The in-game timer pauses.

   Result: PASS

8. FR-GUI-Test-8

   Type: Manual, Functional, Dynamic

   Initial State: The game is in progress, but paused (timer is not running).

   Input: User clicks on the resume game button.

   Expected Output: The in-game timer resumes.

   Output: The in-game timer resumes.

   Result: PASS

9. FR-GUI-Test-9

   Type: Manual, Functional, Dynamic

   Initial State: The game is in progress (timer is running).

   Input: User right clicks on a cell to flag it.

   Expected Output: Game remains in progress and cell becomes flagged.

   Output: Game remains in progress and cell becomes flagged.

   Result: PASS

10. FR-GUI-Test-10

    Type: Manual, Functional, Dynamic

    Initial State: The game is in progress, (timer is running).

    Input: User clicks on the Tidal Blue colour scheme.

    Expected Output: The game board changes colour scheme to Tidal Blue.

    Output: The game board changes colour scheme to Tidal Blue.

    Result: PASS

11. FR-GUI-Test-10

    Type: Manual, Functional, Dynamic

    Initial State: The game is in progress, (timer is running).

    Input: User clicks on the Desert Gold colour scheme.

    Expected Output: The game board changes colour scheme to Desert Gold.

    Output: The game board changes colour scheme to Desert Gold.

    Result: PASS

12. FR-GUI-Test-10

    Type: Manual, Functional, Dynamic

    Initial State: The game is in progress, (timer is running).

    Input: User clicks on the Glacial Blue colour scheme.

    Expected Output: The game board changes colour scheme to Glacial Blue.

    Output: The game board changes colour scheme to Glacial Blue.

    Result: PASS

13. FR-GUI-Test-10

    Type: Manual, Functional, Dynamic

    Initial State: The game is in progress, (timer is running).

    Input: User clicks on the Woodland Green colour scheme.

    Expected Output: The game board changes colour scheme to Woodland Green.

    Output: The game board changes colour scheme to Woodland Green.

    Result: PASS

14. FR-GUI-Test-10

    Type: Manual, Functional, Dynamic

    Initial State: The game is in progress, (timer is running).

    Input: User clicks on the Slate Gray colour scheme.

    Expected Output: The game board changes colour scheme to Slate Gray.

    Output: The game board changes colour scheme to Slate Gray.

    Result: PASS

15. FR-GUI-Test-10

    Type: Manual, Functional, Dynamic

    Initial State: The game is in progress, (timer is running).

    Input: User clicks on the Rose Garden Pink colour scheme.

    Expected Output: The game board changes colour scheme to Rose Garden Pink.

    Output: The game board changes colour scheme to Rose Garden Pink.

    Result: PASS

1.2 Game Structure Testing

1. FR-GS-Test-1

   Type: Structural, Automated, Dynamic

   Initial State: The game is in progress (timer is running).

   Input: Hidden pineapple is clicked upon.

   Expected Output: Game terminates, the timer stops, and the user is unable to click on any other tiles. The locations of all of the hidden mines are revealed. The user is only able to reset the game, change the difficulty using the in-game menu, or close the application.

   Output: Game terminates, the timer stops, and the user is unable to click on any other tiles. The locations of all of the hidden mines are revealed. The user is only able to reset the game, change the difficulty using the in-game menu, or close the application.

   Result: PASS

2. FR-GS-Test-2

   Type: Structural, Automated, Dynamic

   Initial State: The game is in progress (timer is running) with all but one tile containing a hidden pineapple.

   Input: User clicks on the one remaining tile which does not contain a pineapple.

   Expected Output: The game terminates (timer stops). The user must only be able to reset the game, change the difficulty using the in-game menu, or close the application.

   Output: The game terminates (timer stops). The user must only be able to reset the game, change the difficulty using the in-game menu, or close the application.

   Result: PASS

3. FR-GS-Test-3

   Type: Structural, Automated, Dynamic

   Initial State: The game has just commenced (timer is running).

   Input: User clicks on a tile which does not cover a hidden pineapple.

   Expected Output: The tile which is clicked on disappears and the game is still in progress (timer running).

   Output: The tile which is clicked on disappears and the game is still in progress (timer running).

   Result: PASS

2 Nonfunctional Requirements Evaluation

2.1 Usability

1. NFR-USE-Test-1

   Type: Manual, Structural, Static

   Initial State: Game has not yet been executed on the Windows operating system.

   Input: Game has been launched.

   Expected Output: The game functions as expected.

   Output: The game functions as expected.

   Result: PASS

2. NFR-USE-Test-1

   Type: Manual, Structural, Static

   Initial State: Game has not yet been executed on the Mac operating system.

   Input: Game has been launched.

   Expected Output: The game functions as expected.

   Output: The game functions as expected.

   Result: PASS

3. NFR-USE-Test-1

   Type: Manual, Structural, Static

   Initial State: Game has not yet been executed on the Linux operating system.

   Input: Game has been launched.

   Expected Output: The game functions as expected.

   Output: The game functions as expected.

   Result: PASS

2.2 Performance

1. NFR-Perf-Test-1

   Type: Manual, Functional, Dynamic

   Initial State: The game is in progress (timer is running).

   Input: Click on the new game button which produces a response from the game.

   Expected Output: New game starts in under RESPONSETIME.

   Output: New game starts in under RESPONSETIME.

   Result: PASS

2. NFR-Perf-Test-1

   Type: Manual, Functional, Dynamic

   Initial State: The game is in progress (timer is running).

   Input: Click on either the beginner, intermediate, or advanced difficulty button.

   Expected Output: Game dimensions set correspondingly and number of mines set correspondingly within RESPONSETIME.

   Output: Game dimensions set correspondingly and number of mines set correspondingly within RESPONSETIME.

   Result: PASS

3. NFR-Perf-Test-1

   Type: Manual, Functional, Dynamic

   Initial State: The game is in progress (timer is running).

   Input: Click on either the beginner, intermediate, or advanced difficulty button.

   Expected Output: Game dimensions set correspondingly and number of mines set correspondingly within RESPONSETIME.

   Output: Game dimensions set correspondingly and number of mines set correspondingly within RESPONSETIME.

   Result: PASS

4. NFR-Perf-Test-1

   Type: Manual, Functional, Dynamic

   Initial State: The game is in progress (timer is running).

   Input: Click on one of the seven colour scheme buttons.

   Expected Output: Colour scheme is set within RESPONSETIME.

   Output: Colour scheme is set within RESPONSETIME.

   Result: PASS

2.3 Security

1. NFR-Sec-Test-1

   Type: Manual, Structural, Dynamic

   Initial State: The game is in progress (timer is running).

   Input: Give invalid keyboard inputs.

   Expected Output: The game state does not change.

   Output: The game state does not change.

   Result: PASS

2. NFR-Sec-Test-1

   Type: Manual, Structural, Dynamic

   Initial State: The game is paused (timer is not running).

   Input: Give invalid keyboard inputs.

   Expected Output: The game state does not change.

   Output: The game state does not change.

   Result: PASS

3. NFR-Sec-Test-1

   Type: Manual, Structural, Dynamic

   Initial State: The game is paused (timer is not running).

   Input: Click on covered cells.

   Expected Output: Cells remained covered.

   Output: Cells remain covered.

   Result: PASS

4. NFR-Sec-Test-1

   Type: Manual, Structural, Dynamic

   Initial State: The game is paused (timer is not running).

   Input: Flag covered cells.

   Expected Output: Cells remained covered and not flagged.

   Output: Cells remain covered and not flagged.

   Result: PASS

5. NFR-Sec-Test-1

   Type: Manual, Structural, Dynamic

   Initial State: The game is in progress (timer is running).

   Input: Flag uncovered cells.

   Expected Output: Cells remained uncovered and not flagged.

   Output: Cells remain uncovered and not flagged.

   Result: PASS

6. NFR-Sec-Test-1

   Type: Manual, Structural, Dynamic

   Initial State: The game is terminated (timer is not running).

   Input: Click on uncovered cells.

   Expected Output: Cells remain uncovered and game remains terminated.

   Output: Cells remain uncovered and game remains terminated.

   Result: PASS

7. NFR-Sec-Test-1

   Type: Manual, Structural, Dynamic

   Initial State: The game is terminated (timer is not running).

   Input: Flag cells.

   Expected Output: Cells remained uncovered and not flagged. The game state remains terminated.

   Output: Cells remain uncovered and not flagged. The game state remains terminated.

   Result: PASS

3 Unit Testing

1. JUNIT-Test-1

   Type: Automated, Static

   Initial State: Start game with number of mines based on size of the grid.

   Input: BDIMENSIONS

   Expected Output: BMINES

   Output: BMINES

   Result: PASS

2. JUNIT-Test-1 Type: Automated, Static

   Initial State: Start game with number of mines based on size of the grid.

   Input: IDIMENSIONS

   Expected Output: IMINES

   Output: IMINES

   Result: PASS

3. JUNIT-Test-1

   Type: Automated, Static

   Initial State: Start game with number of mines based on size of the grid.

   Input: ADIMENSIONS

   Expected Output: AMINES

   Output: AMINES

   Result: PASS

4. JUNIT-Test-2

   Type: Automated, Static

   Initial State: Start game with beginner difficulty

   Input: Beginner difficulty.

   Expected Output: BDIMENSIONS

   Output: BDIMENSIONS

   Result: PASS

5. JUNIT-Test-2

   Type: Automated, Static

   Initial State: Start game with intermediate difficulty

   Input: Beginner difficulty.

   Expected Output: IDIMENSIONS

   Output: IDIMENSIONS

   Result: PASS

6. JUNIT-Test-2

   Type: Automated, Static

   Initial State: Start game with advanced difficulty

   Input: Beginner difficulty.

   Expected Output: ADIMENSIONS

   Output: ADIMENSIONS

   Result: PASS

4 Changes Due to Testing

The implementation was previously drawing the grid with contents of cell (number, pine mine and empty space) using JPanel draw tools, which would offset the orientation of the grid from the window. While testing disable button functionality for flag, we discovered that the outline of the button resembles the grid revealed when the cell registered a left click. Hence, the implementation was changed such that the button component of cell was assigned an icon to be revealed as it is disabled, based on the content of the button.

On reviewing the results of survey testing, one suggestion given to us was that instead of implementing pause game functionality as it was done in the reference application JSweeper (registering pause and resume when an action was performed outside of game window), we should introduce a pause/resume button for the user to use.

Visual themes Slate Gray and Rose Garden Pink were added based on suggestions made by survey testers.

5 Trace to Requirements

Req.	Tests
FR 1	FR-GUI-Test-1, FR-GUI-Test-2 FR-GUI-Test-1, NFR-Perf-Test-1 JUNIT-Test-1, JUNIT-Test-2
FR 2	FR-GUI-Test-1, FR-GUI-Test-2, JUNIT-Test-1, JUNIT-Test-2
FR 3	FR-GUI-Test-5, FR-GUI-Test-6
FR 4	FR-GUI-Test-5, FR-GUI-Test-6
FR 5	FR-GS-Test-1
FR 6	FR-GUI-Test-6
FR 7	FR-GUI-Test-9
FR 8	FR-GS-Test-1, FR-GS-Test-2
FR 9	FR-GUI-Test-7, FR-GUI-Test-8
FR 10	FR-GS-Test-1
FR 11	FR-GUI-Test-4

Table 2: Trace Between Requirements and Tests

6 Trace to Modules

Req.	**Tests**
M1	FR-GUI-Test-5, FR-GUI-Test-6 FR-GUI-Test-9, FR-GS-Test-1 FR-GS-Test-2
M2	FR-GUI-Test-1, FR-GUI-Test-2 FR-GUI-Test-3
M3	FR-GUI-Test-10
M4	FR-GUI-Test-1, FR-GUI-Test-2, FR-GUI-Test-3, FR-GUI-Test-10
M5	FR-GUI-Test-4, FR-GUI-Test-5, FR-GUI-Test-6, FR-GUI-Test-7, FR-GUI-Test-8, FR-GUI-Test-9, FR-GS-Test-1, FR-GS-Test-1
M6	FR-GUI-Test-1, FR-GUI-Test-2, FR-GUI-Test-3, FR-GUI-Test-5, FR-GUI-Test-6, FR-GUI-Test-7, FR-GUI-Test-8, FR-GUI-Test-9, FR-GUI-Test-10, FR-GS-Test-1 FR-GS-Test-2
M7	FR-GUI-Test-5, FR-GUI-Test-6, FR-GS-Test-1, FR-GS-Test-2
M8	FR-GUI-Test-4, FR-GS-Test-1, FR-GS-Test-2
M9	FR-GUI-Test-6
M10	FR-GUI-Test-4, FR-GUI-Test-5, FR-GUI-Test-6, FR-GUI-Test-9, FR-GS-Test-1, FR-GS-Test-2 FR-GS-Test-3
M11	

Table 3: Trace Between Requirements and Tests

7 Code Coverage Metrics

While testing *PineSweeper*, JUNIT testing was used to test code. The main testing was done to our backend code as the GUI was manually tested. The backend code is mainly contained in the *JSweepGameControl* and *Cell* classes.