Connect Four Game

*Iteration 6: Implementation of bug fixes and enhancements*

# 1 Objectives and Testing Goals

The Testing Goals for the Connect Four game support the following objectives:

* Define the activities required to prepare for Performance Testing.
* Uncover bugs and implement fixes
* Enhancing program performance

While it was wishful to complete all testing categories, the team decided not to pursue Unit Testing, Integration Testing, System Testing, and Regression testing due to time constraints.

# 2 Testing Results

The only test results available are Performance test results, which are shown below.

**2.1 Unit Testing**

N/A

**2.1 Integration Testing**

N/A

**2.1 System Testing**

N/A

**2.1 Performance Testing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Scenario** | **Test Case** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** |
| Build & Run Project | **TC1** | 1. Open the Connect Four project in Eclipse IDE 2. Build the project 3. Run the project | src package  Image package | Project built and ran with Main panel open | Project built and ran in console mode |
| Main panel opens successfully | **TC1** | (1) Open the Connect Four project in Eclipse IDE  (2) Build the project  (3) Run the project | N/A | Eclipse IDE opened the Connect Four project successfully | Eclipse IDE opened the Connect Four project successfully |
| Player 1’s default value | **TC2** | 1. Open the Connect Four project in Eclipse IDE 2. Run the project 3. Click the “**PLAY**” button | N/A | Player 1’s default value is **Player 1** | Player 1’s default value is **Player 1** |
| Player 2’s default value | **TC3** | 1. Open the Connect Four project in Eclipse IDE 2. Run the project 3. Click the “**PLAY**” button | N/A | Player 2’s default value is **Player 2** | Player 2’s default value is **Player 2** |
| Player 2’s value becomes “**Computer**” | **TC4** | 1. Open the Connect Four project in Eclipse IDE 2. Run the project 3. Click the “**Play Against Computer**” button | N/A | Player 2’s default value is **Computer** | Player 2’s default value is **Computer** |
| Main panel transitions to Game Play panel | **TC5** | 1. Open the Connect Four project in Eclipse IDE 2. Run the project 3. Click the “**PLAY**” button | N/A | The **Main** panel transitioned to the **Game Play** panel | The **Main** panel transitioned to the **Game Play** panel |
| Proper player chip position | **TC6** | 1. Open the Connect Four project in Eclipse IDE 2. Run the project 3. Click the “**PLAY**” button 4. Place the colored chips into the empty board array | N/A | The players’ chips dropped into the board from bottom row upward | The players’ chips dropped into the board from bottom row upward |
| Connect Four diagonally | **TC7** | 1. Open the Connect Four project in Eclipse IDE 2. Run the project 3. Click the “**PLAY**” button 4. Win the game by arranging for four chips of the same color to line up linearly and diagonally. | N/A | A winning condition detected. The Game Play panel transitions to the Game Over panel with Winner found as Player 1, or Player 2, or Computer | A winning condition detected. The Game Play panel transitions to the Game Over panel with Winner found as Player 1, or Player 2, or Computer |
| Connect Four horizontally | **TC8** | 1. Open the Connect Four project in Eclipse IDE 2. Run the project 3. Click the “**PLAY**” button 4. Win the game by arranging for four chips of the same color to line up linearly and horizontally. | N/A | A winning condition detected. The Game Play panel transitions to the Game Over panel with Winner found as Player 1, or Player 2, or Computer | A winning condition detected. The Game Play panel transitions to the Game Over panel with Winner found as Player 1, or Player 2, or Computer |
| Connect Four vertically | **TC9** | 1. Open the Connect Four project in Eclipse IDE 2. Run the project 3. Click the “**PLAY**” button 4. Win the game by arranging for four chips of the same color to line up linearly and vertically. | N/A | A winning condition detected. The Game Play panel transitions to the Game Over panel with Winner found as Player 1, or Player 2, or Computer | A winning condition detected. The Game Play panel transitions to the Game Over panel with Winner found as Player 1, or Player 2, or Computer |
| Alternating players’ turns | **TC10**  **TC11**  **TC12**  **TC13** | 1. Open the Connect Four project in Eclipse IDE 2. Run the project 3. Click the “**PLAY**” button 4. Win the game by arranging for four chips of the same color to line up linearly and horizontally, or vertically, or diagonally. | N/A | The players turns alternates when the “**Play Again**” button clicks. | The players turns alternates when the “**Play Again**” button clicks. |
| Dynamic “**Round number**” | **TC14** | 1. Open the Connect Four project in Eclipse IDE 2. Run the project 3. Click the “**PLAY**” button 4. Play the game by dropping circular colored chips onto board. | N/A | The “**Round number**” value is updated every time the players make their move. | The “**Round number**” value is updated every time the players make their move. |
| Win/Lose/Tie logic | **TC15** | 1. Open the Connect Four project in Eclipse IDE 2. Run the project 3. Click the “**PLAY**” button 4. Win the game by arranging for four chips of the same color to line up linearly and horizontally, or vertically, or diagonally. | N/A | The game statistics are updated correctly to reflect players’ winning scores | The game statistics are updated correctly to reflect players’ winning scores |
| Win/Lose/Tie logic | **TC15** | 1. Open the Connect Four project in Eclipse IDE 2. Run the project 3. Click the “**PLAY**” button 4. Play the game and arranging for both players, either human against human or human against computer, to end the game in a tie. | N/A | The Game Play panel transitions to the Game Over panel with a “**tie**” mess | The game cleared the board and threw a null-pointer exception |
| Win/Lose/Tie logic | **TC15**  **TC16** | 1. Open the Connect Four project in Eclipse IDE 2. Run the project 3. Click the “**PLAY**” button 4. Win the game by arranging for four chips of the same color to line up linearly and horizontally, or vertically, or diagonally. | N/A | The game statistics are updated correctly to reflect players’ winning scores.  The Game Play panel paused a few seconds before transitioning to the Game Over panel | The game statistics are updated correctly to reflect players’ winning scores.  The Game Play panel transitioned to the Game Over panel immediately |
| Last panel loops back to first panel | **TC17** | 1. Open the Connect Four project in Eclipse IDE 2. Run the project 3. Click the “PLAY” button 4. Win the game by arranging for four chips of the same color to line up linearly and horizontally, or vertically, or diagonally. 5. Click the “**Main Menu**” button | N/A | The **Game Over** panel transition to the **Main** panel immediately. | The **Game Over** panel transition to the **Main** panel immediately. |
| Panel’s ability to minimize, maximize, and close | **TC18**  **TC19** | 1. Open the Connect Four project in Eclipse IDE 2. Run the project 3. In either Main, or Game Play, or Game Over panel, click on the minimize button, or the maximize button, or the exit button 4. Verify panel behavior | The minimize button | The panel minimizes | The panel minimizes |
| Panel’s ability to minimize, maximize, and close | **TC18**  **TC19** | 1. Open the Connect Four project in Eclipse IDE 2. Run the project 3. In either Main, or Game Play, or Game Over panel, click on the minimize button, or the maximize button, or the exit button 4. Verify panel behavior | The maximize button | The panel maximizes | The panel maximizes |
| Panel’s ability to minimize, maximize, and close | **TC18**  **TC19** | 1. Open the Connect Four project in Eclipse IDE 2. Run the project 3. In either Main, or Game Play, or Game Over panel, click on the minimize button, or the maximize button, or the exit button 4. Verify panel behavior | The exit button | The Connect Four program terminates | The Connect Four program terminates |

**2.1 Regression Testing**

N/A

# 3 Summary of Bug Fixes

Sub-section descriptions of individual bug fixes (one per enhancement) that were implemented. Include operational description of the bug fix, high level software impact of the bug fix to the code base, and amount/type of testing associated with the bug fix. Include screen shots if deemed desired/appropriate by the team.

**4.1 Bug Fix 1**

xxx

**4.2 Bug Fix 2**

xxx

**4.3 Bug Fix 3**

xxx

**4.4 Bug Fix 4**

Xxx

# 4 Summary of Enhancements

Sub-section descriptions of individual enhancements (one per enhancement) that were implemented. Include operational description of the enhancements, high level software impact of the enhancement to the code base, and amount/type of testing associated with the enhancement. Include screen shots if deemed desired/appropriate by the team.

**4.1 Enhancement 1**

xxx

**4.2 Enhancement 2**

xxx

**4.3 Enhancement 3**

xxx

**4.4 Enhancement 4**

xxx

# 5 Significant Challenges

1 to 5 short paragraphs providing an introspective regarding the significant development (technical or management) related challenges associated with the project.

# 6 What We Learned

1 to 5 short paragraphs providing an introspective regarding what individual or team observations were experienced, specifically relating to concepts covered in this course (or the prior software development course (SER 215) in this series). These observations could be from either positive or negative experiences.

# 7 What We Would Improve

1 to 5 short paragraphs providing an introspective regarding team approach or development/administrative activities your team would/might do differently if you were starting over on (this or another) project again. These observations could be from either positive or negative experiences.