**Project Sprint #2**

Implement the following features of the SOS game: (1) the basic components for the game options (board size and game mode) and initial game, and (2) S/O placement for human players ***without*** checking for the formation of SOS or determining the winner. The following is a sample interface. The implementation of a GUI is strongly encouraged. You should practice object-oriented programming, making your code easy to extend. It is important to separate the user interface code and the game logic code into different classes (refer to the TicTacToe example). xUnit tests are required.

|  |  |  |
| --- | --- | --- |
| SOS Icon  Description automatically generated Simple game Icon  Description automatically generated General game Board size  8 | | |
| Blue player  Icon  Description automatically generated S  Icon  Description automatically generated O | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  | | O |  |  |  |  |  |  |  | |  |  | S | O | S |  |  |  | |  |  |  |  | S |  |  |  | |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | S | | Red player  Icon  Description automatically generated S  Icon  Description automatically generated O |
|  | Current turn: blue (or red) |  |

Figure 1. Sample GUI layout of the Sprint 2 program

**Deliverables:**

1. **Demonstration (8 points)**

Submit a video of no more than three minutes, clearly demonstrating that you have implemented the required features and written some automated unit tests. In the video, you must explain what is being demonstrated.

|  |  |  |
| --- | --- | --- |
|  | **Feature** |  |
| 1 | Initial game of the chosen board size and game mode |  |
| 2 | “S” moves |  |
| 3 | “O” moves |  |
| 4 | Unit tests |  |
| 5 | Show whose turn it is |  |
| 6 | Resizing of the board |  |

1. **Summary of Source Code (1 points)**

|  |  |  |
| --- | --- | --- |
| Source code file name | Production code or test code? | # lines of code |
|  |  |  |
| Board.java – 71 lines  Box.java – 131 lines  GameMode.java – 53 lines  GUI.java – 125 lines  HelloApplication.java – 33 lines  HelloController.java – 8 lines  PlayerBox.java – 48 lines | AC1TestCases.java – 41 lines  AC2TestCases.java – 35 lines  AC4TestCases.java – 48 lines  AC6TestCases.java – 47 lines | Source Code:  463 lines  Test Code:  171 lines |
| Total | | 640 lines |

**You must submit all source code to get any credit for this assignment.**

1. **Production Code vs User stories/Acceptance Criteria (3 points)**

Summarize how each user story/acceptance criterion is implemented in your production code (class name and method name etc.)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **User Story ID and Name** | **AC ID** | **Class Name(s)** | **Method Name(s)** | **Status (complete or not)** | **Notes (optional)** |
| 1 | 1.1 | Board | setBoardSize  initBoard  addBox | Complete | setBoardSize sets the board size to the specified size only if it is greater than 2. initBoard and addBox sets up the ArrayList depending on the boardSize. |
|  | 1.1 | GUI | makeBoard | Complete | Function sets the board up according to Board’s specified size (specifically, Board’s boardSize). |
|  | 1.2 | Board | setBoardSize | Complete | Function will not set the boardSize to 2 or anything less than 2. |
| 2 | 2.1 | Board | getIsSimpleGame | Complete | Checks to see if it is a simple game (true) or a general game (false |
|  | 2.1 | GameMode | getGG, getSG | Complete | Gets the value of whether the Simple Game radio button is pressed or the General Game radio button is presses |
| 4 | 4.1 | Box | Box (the constructor)  drawO  drawS | Complete | Allows a ‘S’ or an ‘O’ to be placed in a square. |
|  | 4.1 | GUI | makeBoard | Complete | In the for loop, an nxn grid is created using Boxes. These Boxes are stored in both the ArrayList in Board and in the Pane GameBoard which will be used for the GUI. |
|  | 4.1 | PlayerBox | PlayerBox(constructor)  getS  getO | Complete | Checks to see if the S radio button is clicked or if the O radio button is clicked so it can know which one to place. |
|  | 4.1 | Board | getIsSimpleGame | Complete | Gets the Boolean value that it is a Simple Game. |
|  | 4.2 | Box | Box(the constructor)  drawO  drawS | Complete | Won’t allow a new S or O to be placed in a place that is already filled. |
|  | 4.2 | Board | getIsSimpleGame | Complete | Gets the Boolean value that it is a simple game. |
| 6 | 6.1 | Box | Box (the constructor)  drawO  drawS | Complete | Allows a ‘S’ or an ‘O’ to be placed in a Box. |
|  | 6.1 | GUI | makeBoard | Complete | In the for loop, an nxn grid is created using Boxes. These Boxes are stored in both the ArrayList in Board and in the Pane GameBoard which will be used for the GUI. |
|  | 6.1 | PlayerBox | PlayerBox(constructor)  getS  getO | Complete | Checks to see if the S radio button is clicked or if the O radio button is clicked so it can know which one to place. |
|  | 6.1 | Board | getIsSimpleGame | Complete | Gets the Boolean value that it is a General Game. |
|  | 6.2 | Box | Box(the constructor)  drawO  drawS | Complete | Won’t allow a new S or O to be placed in a place that is already filled. |
|  | 6.2 | Board | getIsSimpleGame | Complete | Gets the Boolean value that it is a General Game |

1. **Tests vs User stories/Acceptance Criteria (3 points)**

Summarize how each user story/acceptance criterion is tested by your test code (class name and method name) or manually performed tests.

4.1 Automated tests directly corresponding to some acceptance criteria

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Story ID and Name** | **Acceptance Criterion ID** | **Class Name (s) of the Test Code** | **Method Name(s) of the Test Code** | **Description of the Test Case (input & expected output)** |
| 1 | 1.1 | AC1TestCase | testStartBoardWithValidSize | Initiates the board and sets the boardSize to a valid size (8), then checks to so see that the board has set the value to 8 (since it is a valid input) and that isBlueTurn is true (should start as Blue Player’s turn). |
|  | 1.2 | AC1TestCase | testStartBoardWithInvalidSize | Initiates the board and sets the boardSize to a valid size (2), then checks to see that board doesn’t set the value to 2 (since it is an illegal value) and that isBlueTurn is true (should start as Blue Player’s turn). |
| 2 | 2.1 | AC2TestCase | testSimpleGameModeSelected | Initiates the board and sets the game mode to a Simple Game. Then checks to see that the game mode is Simple Game. |
|  | 2.1 | AC2TestCase | testGeneralGameModeSelected | Initiates the board and sets the game mode to a General Game. Then checks to see if the game mode is General Game. |
| 4 | 4.1 | AC4TestCase | testSuccesfulMoveInSimpleGame | Sets the board to an 8x8, initiates the board, set the game mode to a Simple Game, and puts an S at (1,1) on the board. Then, checks to see if there is an S at (1,1) and that it is a Simple Game. |
|  | 4.2 | AC4TestCase | testUnsuccessfulMoveInSimpleGame | Sets the board to an 8x8, initiates the board, set the game mode to a Simple Game, and puts an S at (1,1) on the board. It then tries to place an O at (1,1). Then, checks to see if there is an S at (1,1) and that it is a Simple Game. |
| 6 | 6.1 | AC6TestCase | testSuccessfulMoveInGeneralGame | Sets the board to an 8x8, initiates the board, set the game mode to a General Game, and puts an S at (1,1) on the board. Then, checks to see if there is an S at (1,1) and that it is a General Game. |
|  | 6.2 | AC6TestCase | testUnsuccessfulMoveInGeneralGame | Sets the board to an 8x8, initiates the board, set the game mode to a General Game, and puts an S at (1,1) on the board. It then tries to place an O at (1,1). Then, checks to see if there is an S at (1,1) and that it is a General Game. |

4.2 Manual tests directly corresponding to some acceptance criteria

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Story ID and Name** | **Acceptance Criterion ID** | **Test Case Input** | **Test Oracle (Expected Output)** | **Notes** |
| 1 | 1 | 2 in the textField | No change | Not a valid input, so the board shouldn’t change. |
|  | 1 | 4 in the textField | A board size 4x4 | 4 is a valid input, so the board should change |
| 2 | 2.1 | Click on the Simple and General game radio button | Whichever radio button is clicked, that should show as the new selected one. |  |
| 4 | 4.1 | Click the Simple Game button, the O button, and select an open box. | An O should be placed at that box |  |
|  | 4.2 | Click the Simple Game button, the S button, and select an occupied box | An S should not be placed in that box. |  |
| 6 | 6.1 | Click the General Game button, the O button, and select an open box. | An O should be placed at that box |  |
|  | 6.2 | Click the General Game button, the S button, and select an occupied box | An S should not be placed in that box. |  |

4.3 Other automated or manual tests not corresponding to the acceptance criteria

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Test Input** | **Expected Result** | **Class Name of the Test Code** | **Method Name of the Test Code** |
|  |  |  |  |  |
|  |  |  |  |  |