**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 | Choose board size | Fourth test |
| 2 | Choose game mode | First test |
| 3 | Initial game of the chosen board size and game mode | Third test and Second test |
| 4 | “S” moves | Third test and Second test |
| 5 | “O” moves | Third test and Second test |
| 6 | Automated unit tests | All tests are automated |
| … |  |  |

I will upload the video on canvas submission. This is the first time I submit a video if you don’t see any video or could not open the record please let me know so I can send using email.

The order of in record:

1. First test: select game mode and select S or O on each player in each game mode.
2. Second test: Simple game mode first mode testing: select simple game mode. Select board size as 4. Each player will make a move by switching between S and O each turn.
3. Third test: General game mode first mode testing: select General game mode. Select board size as 4. Each player will make a move by switching between S and O each turn.
4. Fourth test: Choose board size. Choose all of the board size from 3 to 9. The default board size will be 0, player have to select board size to start the game.
5. **Summary of Source Code (1 points)**

|  |  |  |
| --- | --- | --- |
| Source code file name | Production code or test code? | # lines of code |
| Main.java | Production | 12 |
| MyFrame.java | Production | 304 |
| MainTest.java | Test code | 213 |
|  | Total | 529 |

**I will submit the source code on canvas. If there are any problem please let me know so I could send it to you by email.**

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

Update your user stories and acceptance criteria from the previous assignment and ensure they adequately capture the requirements. Summarize how each of the following user story/acceptance criteria is implemented in your production code (class name and method name etc.)

|  |  |
| --- | --- |
| **User Story ID** | **User Story Name** |
| 1 | Choose a board size |
| 2 | Choose the game mode of a chosen board |
| 3 | Start a new game of the chosen board size and game mode |
| 4 | Make a move in a simple game |
| 6 | Make a move in a general game |

Location: I put a comment at the head of each Jpanel

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **User Story ID and Name** | **AC ID** | **Class Name(s)** | **Method Name(s)** | **Status (complete or not)** | **Notes (optional)** |
| 1. choose a board size | 1.1 Default size | MyFrame.java  Location: Main/Head frame | Jcombobox gameBoardSize; | Completed | Default will be 0. Player need to choose board size. |
|  | 1.2 choose board size from combo box (3 to 9) | MyFrame.java  Location: Main/Head frame | Jcombobox gameBoardSize;  Integer[] boardSizeSelection={0, 3, 4, 5, 6, 7, 8, 9}; | Completed |  |
| 2. choose the game mode | 2.1 simple game | MyFrame.java  Location: Main/Head frame | JradioButton simpleRadio; | Completed | The default selection |
|  | 2.2 general game | MyFrame.java  Location: Main/Head frame | JradioButton generalRadio; | Completed |  |
| 3 make a move in a simple game | 4.1 making a move simple game | MyFrame.java  Location: actionPerformed | actionPerformed(Actionevent e) | Completed | The making move in 2 game mode are similar. What I’m doing is adding a function name winning condition. In the winning condition if the simple game is selected, the first player who make SOS will win, game end. Else, both player will continue to play until the board is fill, who make the most SOS will be the winner. |
|  | 4.2 switching turn | MyFrame.java  Location: actionPerformed | actionPerformed(Actionevent e) | completed |  |
| 4 make a move in general game | 6.1 making a move in general game | MyFrame.java  Location: actionPerformed | actionPerformed(Actionevent e) | Completed | See make a move in a simple game |
|  | 6.2 switching turn | MyFrame.java  Location: actionPerformed | actionPerformed(Actionevent e) | Completed | See make a move in a simple game |

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

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

|  |  |
| --- | --- |
| **User Story ID** | **User Story Name** |
| 1 | Choose a board size |
| 2 | Choose the game mode of a chosen board |
| 3 | Start a new game of the chosen board size and game mode |
| 4 | Make a move in a simple game |
| 6 | Make a move in a general game |

4.1 Automated tests directly corresponding to the acceptance criteria of the above user stories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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)** |
| 1choose a borad size | 1.1 Check size | MainTest.Java | comboboxTesting() | Ensure the default size will be 0. Select all the size of the available in the combo box with the expect output as a number of the square in the board.  Player turn will display under the board |
| 2 Choose game mode | 2.1 Simple mode | MainTest.Java | gameModeTest() | Ensure the player can select simple mode. Ensure players can select S or O move during that game mode |
|  | 2.2 General mode | MainTest.Java | gameModeTest() | Ensure the player can select general mode. Ensure players can select S or O move during that game mode |
| 3 -> 4 making move | Making move | MainTest.Java | simpleGameModeTest()  generalGameModeTest() | Ensure that players can make the first mode in both game mode depend who turn is it.  Ensure players can use S and O move.  Ensure players can’t select the square which has a letter it, if they select it they have to choose difference square which doesn’t have a letter in other for the turn to switch.  Ensure each player turn display correctly. |

4.2 Manual tests directly corresponding to the acceptance criteria of the above user stories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Story ID and Name** | **Acceptance Criterion ID** | **Test Case Input** | **Test Oracle (Expected Output)** | **Notes** |
| 1. choose a board size | 1.1 Default size | The size when the game is launch is 0 | No board display |  |
|  | 1.2 Select the board size form the combo box (3 to 9) | Select any number from 3 to 9 | The game board will generate depend of the selection |  |
| 2. choose the game mode | 2.1 simple game | Click on simple game radio button | The winning condition for simple mode will apply at the back end |  |
|  | 2.2 general game | Click on general game radio button | The winning condition for general mode will apply at the back end |  |
| 3 make a move in a simple game | 4.1 making a move simple game | the player chooses either S or O and then place it on an empty square | A selected S or O will be placed at a selected square |  |
|  | 4.2 switching turn | Player 1 made a move (using either S or O) | The turn will be switched to player 2 automatically | Who can make a move first will generate randomly at the backend |
| 4 make a move in general game | 6.1 making a move in general game | the player chooses either S or O and then place it on an empty square | A selected S or O will be placed at a selected square |  |
|  | 6.2 switching turn | Player 1 made a move (using either S or O) | The turn will be switched to player 2 automatically | Who can make a move first will generate randomly at the backend |