**Project Sprint #2**

The SOS game is described in CS449HomeworkOverview.docx. You should read the description very carefully.

Your submission must include the GitHub link to your project and you must ensure that the instructor has the proper access to your project. You will receive no points otherwise.

**GitHub link: https://github.com/gwb9h3/SOS-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 link to 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. No points will be given without a video link.

**YouTube/Panopto link: https://umsystem.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=f4d5a055-bd1b-4340-a38c-b20900344b93**

|  |  |  |
| --- | --- | --- |
|  | **Feature** |  |
| 1 | Choose board size |  |
| 2 | Choose game mode |  |
| 3 | Initial game of the chosen board size and game mode |  |
| 4 | “S” moves |  |
| 5 | “O” moves |  |
| 6 | Automated unit tests |  |

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

|  |  |  |
| --- | --- | --- |
| Source code file name | Production code or test code? | # lines of code |
| GameView.java | Production | 181 |
| GameController.java | Production | 85 |
| Main.java | Production | 15 |
| GameControllerTest.java | Test Code | 65 |
| GameModeSelecitonTest.java | Test Code | 40 |
| Total | | 386 |

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

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 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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 | GameView & Game Controller | Stored under the gameView.getBtnEnter action listener | Complete | More classes will be added to store the buttons data in an array when it is necessary to find SOS points. Therefore this code may get moved to a different file and just be called inside this class at that point. |
|  | 1.2 | GameView & Game Controller | Stored under the gameView.getBtnEnter action listener | Complete | Same as previous |
| 2. Choose a game mode | 2.1 | GameView & Game Controller | Stored as radio buttons in the top pannel of GameView and functionality is found in the the gameView.getBtnEnter action listener | Semi-Complete | The game will not start unless the game mode is chosen but the rules for the game mode have not been implemented as the SOS points have not been implemented |
|  | 2.2 | GameView & Game Controller | Stored as radio buttons in the top pannel of GameView and functionality is found in the the gameView.getBtnEnter action listener | Semi-Complete | The game will not start unless the game mode is chosen but the rules for the game mode have not been implemented as the SOS points have not been implemented |
| 3. Start a new game of the chosen board size and game mode | 3.1 | GameView & Game Controller | In gameView the enter button starts it and the code is stored in the the gameView.getBtnEnter action listener | Complete | The game starts and the board is initialized when started but the backend points trackers and rules need to be implemented when the SOS trackers are required |
|  | 3.2 | GameView & Game Controller | In gameView the enter button starts it and the code is stored in the the gameView.getBtnEnter action listener | Complete | The game starts and the board is initialized when started but the backend points trackers and rules need to be implemented when the SOS trackers are required |
| 4. Make a move in a simple game | 4.1 | Game Controller | The action is performed in the gameView.getBtnEnter action listener | Semi-Complete | Move record will be kept once the matrix code is created in a separate file. Placement works perfectly fine though |
|  | 4.2 | Game Controller | The action is performed in the gameView.getBtnEnter action listener | Complete | Invalid moves cannot be done because the button to place a S or O does not work once an S or O is already in a tile. |
|  | 4.3 | Game Controller | The action is performed in the gameView.getBtnEnter action listener | Complete | Turn switching is fully functional and is shown below the board |
|  | 4.4 | Game Controller | The action is performed in the gameView.getBtnEnter action listener | Not Complete | The board still allows for moves to be done once a SOS is formed due to the points not yet being tracked in the game |
| 5. Make a move in a general game | 6.1 | Game Controller | The action is performed in the gameView.getBtnEnter action listener | Semi-Complete | Move record will be kept once the matrix code is created in a separate file. Placement works perfectly fine though |
|  | 6.2 | Game Controller | The action is performed in the gameView.getBtnEnter action listener | Complete | Invalid moves cannot be done because the button to place a S or O does not work once an S or O is already in a tile. |
|  | 6.3 | Game Controller | The action is performed in the gameView.getBtnEnter action listener | Complete | Turn switching is fully functional and is shown below the board |
|  | 66.4 | Game Controller | The action is performed in the gameView.getBtnEnter action listener | Complete | The board does not allow for more moves once a general game is complete because all of the tiles allow for only one move to be done before becoming solid. |

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

You are required to use ChatGPT to create at least 2 unit tests using ChatGPT. You also need to ensure that that the generated user stories are correct, and refined them if not. At the end of the submission, provide the screenshots of your chatgpt prompts and answers, along with errors chatgpt made and you had to correct. You may also use LLMs hosted locally. Points will be deducted if no screenshots provided.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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. Choose a board size | 1.1 | GameControllerTest.java | testInvalidBoardSizeInputOutOfRange() | Test inputs a board size of 2 and it should return null |
|  | 1.2 | GameControllerTest.java | testValidBoardSizeInput() | Test inputs a board size of 5 and should return a list of 25 buttons |
| 2 | 2.1 | GameModeSelectionTest.java | testSimpleGameModeSelected() | Test imitates selecting the simple game mode radio button and checks the getter function to see if it is reflecting being selected |
|  | 2.2 | GameModeSelectionTest.java | testGeneralGameModeSelected() | Test imitates selecting the general game mode radio button and checks the getter function to see if it is reflecting being selected  (HAD TO BE EDITED MANUALLY DUE TO CHANGES IN SOURCE CODE) |

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** |
| 3. Start a new game of the chosen board size and game mode | 3.1 Trying to start the game with invalid board size | Input board size of 2 | Game does not initialize a board and sends an error message |  |
|  | 3.2 Trying to start the game with a valid board size and invalid game selection | Board size of 4 selected but no game mode selected | The game does not initialize a board and sends an error message |  |
|  | 3.3 Starting a game with a valid game mode and valid board size | Game mode of Simple or general and board size  3<=x<=10 | The game initializes the board of given size and game mode |  |
| 4. Making a move in a simple game | 4.1 Valid Move Placement | Place an S or O in an empty tile on the board | The game will put the S or O in that tile |  |
|  | 4.2 Invalid Move Handling | Attempt to place an S or O in a full tile on the board | The game will not allow the button to be pressed to fill the tile as it is already full |  |
|  | 4.3 Turn switch after valid move | After a valid move from either player | The game will switch the text at the bottom and give control to the opposite player |  |
| 6. Making a move in a general game | 6.1 Valid Move Placement | Place an S or O in an empty tile on the board | The game will put the S or O in that tile |  |
|  | 6.2 Invalid Move Handling | Attempt to place an S or O in a full tile on the board | The game will not allow the button to be pressed to fill the tile as it is already full |  |
|  | 6.3 Turn switch after valid move | After a valid move from either player | The game will switch the text at the bottom and give control to the opposite player |  |
|  | 6.4 Prevent further moves after a board is full | After all tiles have filled up on the board | The game will no longer allow any new moves to be placed |  |
|  |  |  |  |  |

4.3 Other automated or manual tests not corresponding to the acceptance criteria of the above user stories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Test Input** | **Expected Result** | **Class Name of the Test Code** | **Method Name of the Test Code** |
| 1, Input is non-numerical | abc | null | GameControllerTest.java | testInvalidBoardSizeInputNonNumeric() |
|  |  |  |  |  |

A black and white screen with white text

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a black and white screen

Description automatically generated

A black screen with white text

Description automatically generatedA screenshot of a computer

Description automatically generated