**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/Pavan698/sprint2**](https://github.com/Pavan698/sprint2)

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 required. You should practice object-oriented programming, making your code easy to extend. It is required 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:**

|  |  |  |
| --- | --- | --- |
|  | **Feature** |  |
| 1 | Choose board size | Only numbers are allowed  Only size >2 is allowed |
| 2 | Choose game mode | We can select simple game or general game |
| 3 | Start a new game of the chosen board size and game mode | If we press on ‘NEW GAME’, a new game with chosen valid board size and mode will be started.  If either board size or game mode is changed any time during a game, a new game must be started by clicking on the ‘New Game’ button. |
| 4 | “S” moves | Player can select the S radio button at his side and click on an empty square to make a S move during his turn. Move cannot be made on an already filled square. |
| 5 | “O” moves | Player can select the O radio button at his/her side to make a O move during his turn. Move cannot be made on an already filled square. |
| 6 | Automated unit tests | testChooseBoardSize testBoardSizeEnteredIsNumeric testBoardSizeEnteredIsGreaterThanTwo testSimpleGameMode testGeneralGameMode testStartGameWithChosenSizeAndMode testMakeSMoveInSimpleGame testMakeOMoveInSimpleGame testMakeSMoveInGeneralGame testMakeOMoveInGeneralGame  testInitialBoardSize testInitializeBoard testInitialPlayerTurn testInitialGameMode testResetGame |

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

|  |  |  |
| --- | --- | --- |
| Source code file name | Production code or test code? | # lines of code |
| Main | Production code | 6 |
| Game | Production code | 63 |
| GUI | Production code | 251 |
| GameTest | Test code | 110 |
| Total | | 430 |

**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)** |
| Choose a board size | 1.1 | GUI  Game | boolean handleBoardSizeTextChange(String txtBoardSize)  public void setBoardSize(int boardSize) | Complete |  |
|  | 1.2 | GUI  Game | boolean handleBoardSizeTextChange(String txtBoardSize)  boolean isBoardSizeTextNumeric(String txtBoardSize)  boolean isBoardSizeGreaterThanTwo(int newSize) | Complete |  |
|  | … |  |  |  |  |
| Choose the game mode of a chosen board | 2.1 | GUI  Game | public void itemStateChanged(ItemEvent e)  public void setSimpleGame(boolean simpleGame) | Complete |  |
|  | 2.2 | GUI  Game | public void itemStateChanged(ItemEvent e)  public void setSimpleGame(boolean simpleGame) | Complete |  |
| Start a new game of the chosen board size and game mode | 3.1 | GUI  Game | private void newGameButtonClicked()  public void reset(int boardSize, boolean isSimpleGame) | Complete |  |
| Make a move in a simple game | 4.1 | GUI  Game | private void cellClicked(JButton button, int finalI, int finalJ)  public void makeMove(int row, int col, String symbol) | Complete |  |
|  | 4.2 | Game | public void setBluePlayersTurn(boolean bluePlayersTurn) | Incomplete | It does not check if a SOS sequence is created before turn alteration yet |
| Make a move in a general game | 6.1 | GUI  Game | private void cellClicked(JButton  button, int finalI, int finalJ)  public void makeMove(int row, int col, String symbol) | Complete |  |
|  | 6.2 | Game | public void setBluePlayersTurn(boolean bluePlayersTurn) | Incomplete | It does not check if a SOS sequence is created before turn alteration yet |

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. You also need to ensure that the generated user stories are correct, and refine 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 fixed. You may also use another LLM, including hosted locally. Points will be deducted if no screenshots are 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)** |
| Choose a board size | 1.1 | GameTest | testChooseBoardSize() | Test setting and getting the board size, expecting the set size to match the get size.  Input: Set board size to 10. Expected Output: Get board size should return 10 |
|  | 1.2 | GameTest | testBoardSizeEnteredIsNumeric | Test checking if a string is numeric, expecting false for a non-numeric string and true for a numeric string.  Input: "a". Expected Output: isBoardSizeTextNumeric should return false. Input: "5". Expected Output: isBoardSizeTextNumeric should return true  Test checking if an entered number is greater than 2, expecting false for a number less than 2 and true for a number greater than 2 |
|  | … |  |  |  |
| Choose the game mode of a chosen board | 2.1 | GameTest | testSimpleGameMode | Test setting and getting the game mode as simple, expecting true for isSimpleGame after setting  Input: Set game mode to simple. Expected Output: isSimpleGame should return true after setting |
|  | 2.2 | GameTest | testGeneralGameMode | Test setting and getting the game mode as general, expecting false for isSimpleGame after setting  Input: Set game mode to general. Expected Output: isSimpleGame should returnfalse after setting. |
| Start a new game of the chosen board size and game mode | 3.1 | GameTest | testStartGameWithChosenSizeAndMode | Test resetting the game with a chosen board size and game mode, expecting the board size and game mode to match the input values.  Input: Board size = 9, isSimpleGame = false. Expected Output: Get board size should return 9, isSimpleGame should return false, and the initial game board should have an empty cell at [0][0]. |
| Make a move in a simple game | 4.1 | GameTest | testMakeSMoveInSimpleGame | Test making an 'S' move in a simple game, expecting the board to have 'S' at the specified position.  Input: Reset game with board size 8 and simple game mode, then make an 'S' move at [5][0]. Expected Output: The board should have 'S' at [5][0]. |
|  | 4.2 | GameTest | testMakeOMoveInSimpleGame | Test making an 'O' move in a simple game, expecting the board to have 'O' at the specified position.  Input: Reset game with board size 8 and simple game mode, then make an 'O' move at [0][2].  Expected Output: The board should have 'O' at [0][2]. |
| Make a move in a general game | 6.1 | GameTest | testMakeSMoveInGeneralGame | Test making an 'S' move in a general game, expecting the board to have 'S' at the specified position.  Input: Reset game with board size 8 and general game mode, then make an 'S' move at [5][0]. Expected Output: The board should have 'S' at [5][0]. |
|  | 6.2 | GameTest | testMakeOMoveInGeneralGame | Test making an 'O' move in a general game, expecting the board to have 'O' at the specified position.  Input: Reset game with board size 8 and general game mode, then make an 'O' move at [0][2]. Expected Output: The board should have 'O' at [0][2]. |

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** |
| Choose a board size | 1.1 | Set board size to 10 | A new 10\*10 game board is displayed | Passed |
|  | 1.2 | Input: “a”  Input: “5” | Error message indicating invalid input  Error message indicating only numbers can be entered for board size. | Passed  Passed |
| Choose the game mode of a chosen board | 2.1 | Set game mode to simple. | Simple Game radio button is selected and General Game radio button deselected | Passed |
|  | 2.2 | Set game mode to general | Simple Game radio button is deselected and General Game radio button selected | Passed |
| Start a new game of the chosen board size and game mode | 3.1 | Board size = 9, Game mode = General Game  Board size=8,  Game mode = Simple game | When ‘New Game’ button is pressed a message indicating the new game start is shown and players are allowed to make moves in a General game of 9\*9 game board.  When ‘New Game’ button is pressed a message indicating the new game start is shown and players are allowed to make moves in a Simple game of 8\*8 game board. | Passed  Passed |
| Make a move in a simple game | 4.1 | During player’s turn in a Simple Game, player selects ‘S’ from his side of radio button and clicks on an empty square  During player’s turn in a Simple Game, player selects ‘O’ from his side of radio button and clicks on an empty square  During player’s turn in a Simple Game, player selects ‘S’ from his side of radio button and clicks on a non-empty square  During a player’s turn in a simple game, player selects ‘O’ from his side of radio button and clicks on a non-empty square | The square which was clicked on is now marked as ‘S’  The square which was clicked on is now marked as ‘O’  An error message is displayed showing player should click on an empty square.  An error message is displayed showing player should click on an empty square. | Passed  Passed  Passed  Passed |
|  | 4.2 | During player’s turn in a Simple Game, player selects ‘S’ from his side of radio  button and clicks on an empty square  During player’s turn in a Simple Game, player selects ‘O’ from his side of radio  button and clicks on an empty square  During player’s turn in a Simple Game, player selects ‘S’ from his side of radio  button and clicks on a non-empty square  During player’s turn in a Simple Game, player selects ‘O’ from his side of radio  button and clicks on a non-empty square | The square which was clicked on is now marked as ‘S’ and turn is altered  The square which was clicked on is now marked as ‘O’ and turn is altered  An error message is displayed showing player should click on an empty square and turn is not altered  An error message is displayed showing player should click on an empty square and turn is not altered | Passed  Passed  Passed  Passed |
| Make a move in a general game | 6.1 | During player’s turn in a General Game, player selects ‘S’ from his side of radio  button and clicks on an empty square  During player’s turn in a General Game, player selects ‘O’ from his side of radio  button and clicks on an empty square  During player’s turn in a General Game, players selects ‘S’ from his side of radio  button and clicks on a non-empty square  During player’s turn in a General Game, player selects ‘O’ from his side of radio  button and clicks on a non-empty square | The square which was clicked on is now marked as ‘S’  The square which was clicked on is now marked as ‘O’  An error message is displayed showing player should click on an empty square.  An error message is displayed showing player should click on an empty square | Passed  Passed  Passed  Passed |
|  | 6.2 | During player’s turn in a General Game, player selects ‘S’ from his side of radio button and clicks on an empty square  During player’s turn in a General Game, player selects ‘O’ from his side of radio button and clicks on an empty square  During player’s turn in a General Game, player selects ‘S’ from his side of radio button and clicks on a non-empty square  During player’s turn in a General Game, player selects ‘O’ from his side of radio button and clicks on a non empty square. | The square which was clicked on is now marked as ‘S’ and turn is altered  The square which was clicked on is now marked as ‘O’ and turn is altered  An error message is displayed showing player should click on an empty square and turn is not altered  An error message is displayed showing player should click on an empty square and turn is not altered. | Passed  Passed  Passed  Passed |

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 | - | Initial board size is 8 | GameTest | testInitialBoardSize |
| 2 | - | Game board is created and its empty | GameTest | testInitializeBoard |
| 3 | - | Initial Player is blue player | GameTest | testInitialPlayerTurn |
| 4 | - | Initial game mode is simple game | GameTest | testInitialGameMode |
| 5 | - | Make a move in this initial game and reset a new General game with a 10\*10 game board. | Game board size is 10. Game mode is not Simple game. A game board is created and its empty | testResetGame |

Screenshots of the ChatGPT generating couple of tests.

A screenshot of a chat

AI-generated content may be incorrect.A screenshot of a computer game

AI-generated content may be incorrect.A screenshot of a chat

AI-generated content may be incorrect.A screenshot of a chat

AI-generated content may be incorrect.