**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:**

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 |
| 2 | Choose game mode |
| 3 | Start a new 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 |
| main.py | Production | 36 |
| sos\_game.py | production | 39 |
| sos\_gui.py | production | 68 |
| test\_board\_size.py | test | 33 |
| test\_game\_mode.py | test | 30 |
| test\_general\_game.py | test | 31 |
| test\_new\_game.py | test | 29 |
| test\_simple\_game.py | test | 49 |
|  | **Total** | 315 |

**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 | main.py, SOSGame | \_\_init\_\_() | complete |  |
|  | 1.2 | main.py, SOSGUI | launchGame(), setup\_ui() | complete |  |
|  | 1.3 | main.py, SOSGame | get\_user\_input(), \_\_init\_\_() | complete | AI generatted |
|  |  |  |  |  |  |
| 2. Choose the game mode | 2.1 | main.py, SimpleSOS, GeneralSOS | launchGame() | complete |  |
|  | 2.2 | SimpleSOS, GeneralSOS | \_\_init\_\_() | complete |  |
|  |  |  |  |  |  |
| 3. Start a new game | 3.1 | SOSGUI, SOSGame | restart\_game() | complete |  |
|  |  |  |  |  |  |
| 4. Make a move in a simple game | 4.1 | SimpleSOS, SOSGame | makeMove() | complete |  |
|  | 4.2 | SimpleSOS, SOSGame | makeMove() | complete |  |
|  | 4.3 | SimpleSOS, SOSGame | makeMove() | complete | AI generated |
|  |  |  |  |  |  |
| 6. Make a move in a general game | 6.1 | GeneralSOS, SOSGame | makeMove() | complete |  |
|  | 6.2 | GeneralSOS, SOSGame | makeMove() | complete |  |
|  | 6.3 | GeneralSOS, SOSGame | makeMove() | complete |  |

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)** |
| 1. Choose a board size | 1.1 | TestBoardSize | test\_valid\_board\_size() | Input: SOSGame(4), Expected: 4x4 board is created. |
|  | 1.2 | TestBoardSize | test\_correct\_board\_size() | Input: Different sizes (3,5,7), Expected: Board matches size. |
|  | 1.3 | TestBoardSize | test\_small\_board\_size() (AI generated) | Input: SOSGame(2), Expected: Raises ValueError. |
|  |  |  |  |  |
| 2. Choose the game mode | 2.1 | TestGameMode | test\_simple\_mode() | Input: SimpleSOS(3), Expected: Instance of SimpleSOS. |
|  | 2.2 | TestGameMode | test\_general\_mode() | Input: GeneralSOS(3), Expected: Instance of GeneralSOS. |
|  |  |  |  |  |
| 3. Start a new game | 3.1 | TestNewGame | test\_board\_reset() | Input: Restart game, Expected: Board is cleared. |
|  |  |  |  |  |
| 4. Make a move in a simple game | 4.1 | TestSimpleGam | test\_valid\_move() | Input: Move at (1,1), Expected: Board updates, turn switches. |
|  | 4.2 | TestSimpleGam | test\_occupied\_cell\_move() | Input: Move at (1,1) twice, Expected: Second move fails. |
|  | 4.3 | TestSimpleGam | test\_out\_of\_bounds\_move() (AI-generated) | Input: Move outside board, Expected: Move fails, turn does not switch. |
|  |  |  |  |  |
| 6. Make a move in a general game | 6.1 | TestGeneralGame | test\_valid\_move() | Input: Move at (1,1), Expected: Board updates, turn switches. |
|  | 6.2 | TestGeneralGame | test\_occupied\_cell() | Input: Move at (0,0) twice, Expected: Second move fails. |
|  | 6.3 | TestGeneralGame | test\_out\_of\_bounds\_move() | Input: Move outside board, Expected: Move fails, turn does not switch. |
|  |  |  |  |  |
|  |  |  |  |  |

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 | Player inputs board size 4 | Board updates to 4x4 grid |  |
|  | 1.2 | Player confirms board size | Board displays correct size |  |
|  |  |  |  |  |
| 2. Choose the game mode | 2.1 | Player selects Simple mode | Game starts in SimpleSOS mode |  |
|  | 2.2 | Player selects General mode | Game starts in GeneralSOS mode |  |
|  |  |  |  |  |
| 3. Start a new game | 3.1 | Player clicks "New Game" | Game starts in GeneralSOS mode |  |

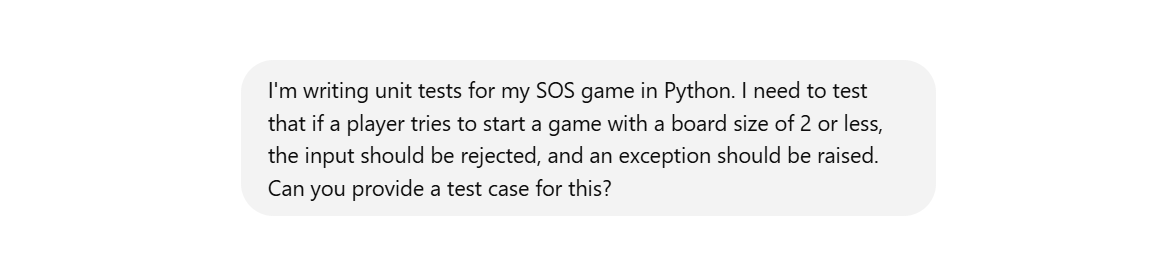
4.3 Other automated or manual tests not corresponding to the acceptance criteria of the above user stories

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



**AI Prompts & Code (Screenshots)**

**AC 1.3:**



**AC 4.3:**

