**Project Sprint #1**

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-1

In this assignment, you aim to specify the requirements (i.e., user stories and acceptance criteria) of the target software that allows a human player to play a simple or general SOS game against a human opponent. These requirements will be fully implemented by the end of sprint 3. The minimum features include **choosing the board size,** **choosing the game mode (simple or general)**, **starting a new game**, **making a move (in a simple or general game)**, **determining if a simple or general game is over**. The following is a sample GUI layout.

|  |  |  |
| --- | --- | --- |
| 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 | Chart, line chart  Description automatically generated | Red player  Icon  Description automatically generated S  Icon  Description automatically generated O |
|  | Current turn: blue (or red) | New Game |

Figure 1. Sample GUI layout of the first working program by the end of Sprint 3

Use the following tables to document your user stories and acceptance criteria.

You are required to use the free ChatGPT version to complete 2 user stories and their respective acceptance criteria. You also need to ensure that the generated user stories are correct and refine them if not. At the end of the submission, provide screenshots of your ChatGPT prompts and answers, along with errors the ChatGPT made and that you had to correct. You may also use LLMs hosted locally. Points will be deducted if no screenshots are provided.

1. **User Stories (3 points)**

* **User Story Template**: As a <role>, I want <goal> [so that <benefit>]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **User Story Name** | **User Story Description** | **Priority** | **Estimated effort (hours)** |
| 1 | Choose a board size | As a player, I want to chose a board size so that I can tailor the game complexity to my liking. | 1 | 2 |
| 2 | Choose the game mode of a chosen board | As a player, I want to be able to pick from multiple game modes for improved replayability. | 2 | .5 |
| 3 | Start a new game of the chosen board size and game mode | As a player, I want to start a new game with the chosen board size and game mode so that I can customize the game experience to my preferences. | 3 | 1 |
| 4 | Make a move in a simple game | As a player, I want to make a move in a simple game so that I can try to form an SOS and win the game. | 4 | 1 |
| 5 | A simple game is over | As a player, I would like the system to recognize when a simple game is over to let the user know if they won or lost. | 6 | 2 |
| 6 | Make a move in a general game | As a player, I want to | 5 | 1 |
| 7 | A general game is over | As a player, I would like the system to recognize when a general game is over to let the user know if they won or lost. | 7 | 2.5 |

1. **Acceptance Criteria (AC) (8 points): Add/delete rows as needed.**

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story ID and Name** | **AC**  **ID** | **Description of Acceptance Criterion** | **Status (completed, toDo, inPprogress)** |
| 1. Choose a board size | 1.1 | **AC 1.1 User Inputs Invalid Board Size**  Given: the player has selected a board size less than 3 tiles wide,  When: the player chooses to start a new game,  Then: the game should display an error message prompting the player to choose a proper board size. | toDO |
| 1.2 | **AC 1.2 User Inputs a Proper Board Size**  Given: the player has selected a board size greater than 2 tiles wide,  When: the player chooses to start a new game,  Then: the game should display a board of the given NxN size. | toDO |
| 2. Choose the game mode of a chosen board | 2.1 | **AC 2.1 Simple Game is Chosen**  Given: the player has selected to play a simple game,  When: the player chooses to start a new game,  Then: the rules will be set to the first player to get a SOS wins. | toDO |
| 2.2 | **AC 2.2 General Game is Chosen**  Given: the player has selected to play a general game,  When: the player chooses to start a new game,  Then: the rules will be set to the player with the most SOS’s wins. | toDO |
| 3. Start a new game of the chosen board size and game mode | 3.1 | **AC 3.1: Game Setup with Chosen Board Size**  **Given:** the player has selected a specific board size,  **When:** the player chooses to start a new game,  **Then:** the game should initialize with the selected board size and display an empty game board. | toDO |
| 3.2 | **AC 3.2: Game Setup with Simple Game Mode**  **Given:** the player has selected the “Simple Game” mode, **When:** the player chooses to start a new game, **Then:** the game should initialize in the “Simple Game” mode with the chosen board size. | toDO |
| 3.3 | **AC 3.3: Game Setup with General Game Mode**  **Given:** the player has selected the “General Game” mode, **When:** the player chooses to start a new game, **Then:** the game should initialize in the “General Game” mode with the chosen board size. | toDO |
| 3.4 | **AC 3.4: Error Handling for Missing Selection**  **Given:** the player has not selected a board size or a game mode, **When:** the player tries to start a new game, **Then:** the game should display an error message prompting the player to choose both a board size and a game mode. | toDO |
| 3.5 | **AC 3.5: Confirmation of Game Start**  **Given:** the player has selected both a valid board size and a game mode, **When:** the player starts the new game, **Then:** a confirmation message should be displayed indicating the selected settings (board size and game mode), and the game should begin with an empty board. | toDO |
| 4. Make a move in a simple game | 4.1 | **AC 4.1: Valid Move Placement**  **Given:** the player selects an empty cell on the board, **When:** the player places an "S" or "O" in the cell, **Then:** the chosen letter should appear in the cell and the move should be recorded. | toDO |
| 4.2 | **AC 4.2: Invalid Move Handling**  **Given:** the player selects an occupied cell, **When:** the player attempts to place an "S" or "O" in that cell, **Then:** an error message should be displayed indicating that the cell is already occupied, and the player should be prompted to select an empty cell. | toDO |
| 4.3 | **AC 4.3: Turn Switch After Valid Move**  **Given:** a player has made a valid move, **When:** the move is completed, **Then:** the turn should switch to the other player, and they should be prompted to make their move. | toDO |
| 4.4 | **AC 4.4: Prevent Further Moves After Game Over**  **Given:** the game is marked as over (by another condition or user story), **When:** a player attempts to make a move, **Then:** the move should be prevented, and a message should indicate that the game has ended. | toDO |
| 5. A simple game is over | 5.1 | **AC 5.1: A Player has Gotten SOS** **Given:** a player has made a valid move creating an SOS, **When:** the move goes through, **Then:** the game will recognize the player who created the SOS as the winner. | toDO |
| 5.2 | **AC 5.2: The Board is Full** **Given:** each cell on the board is full and there is no SOS on the board,  **When:** the game recognizes the board is full, **Then:** the game will declare that the game is a tie. | toDO |
| 6. Make a move in a general game | 6.1 | **AC 6.1: Valid Move Placement Given:** the player selects an empty cell on the board, **When:** the player places an "S" or "O" in the cell, **Then:** the chosen letter should appear in the cell and the move should be recorded. | toDO |
| 6.2 | **AC 6.2: Invalid Move Handling**  **Given:** the player selects an occupied cell, **When:** the player attempts to place an "S" or "O" in that cell, **Then:** an error message should be displayed indicating that the cell is already occupied, and the player should be prompted to select an empty cell. | toDO |
| 6.3 | **AC 6.3: Turn Switch After Valid Move**  **Given:** a player has made a valid move, **When:** the move is completed, **Then:** the turn should switch to the other player, and they should be prompted to make their move. | toDO |
| 6.4 | **AC 6.4: Prevent Further Moves After Board is Full**  **Given:** the board has filled up, **When:** a player attempts to make a move, **Then:** the move should be prevented, and a message should indicate that the game has ended. | toDO |
| 7. A general game is over | 7.1 | **AC 7.1: The Board is Full Given:** the board has filled up,  **When:** the game moves on to count all of the SOS’s on the board, **Then:** the game will announce the player with the most SOS’s as the winner. | toDO |
| 7.2 | **AC 7.2: The Game is a Tie**  **Given:** the game has ended and both players have the same number of SOS’s, **When:** the game goes to announce the winner, **Then:** the game will announce that the game is a tie. | toDO |

1. **Data flow diagram (DFD) (4 points):**

Consider a web-based SOS game that allows players from all over the world to play SOS games against each other (similar to chess.com for chess). Using data flow digram examples presented in class, draw the data flow diagram for your global SOS game. You may use the tool of your choice, but it is a good idea to get familiar with draw.io (https://app.diagrams.net/) or other similar tools

A screenshot of a black screen

Description automatically generated