**Sprint 1**

**User Stories**

| **ID** | **User Story Name** | **User Story Description** | **Priority** | **Estimated Effort(Hours)** |
| --- | --- | --- | --- | --- |
| 1 | Choose a board size | As a player, I want to choose a board size, so I can start a game | 1 | 1 |
| 2 | Choose the game mode of a chosen board | As a player, I want to choose a game mode, so I can start a game | 1 | 1 |
| 3 | Start a new game of the chosen board size and game mode | As a player, I want to start a game with the desired options, so I can play the game type I have chosen | 2 | 3 |
| 4 | Make a move in a simple game | As a player, I want to make a move in a simple game so the other player can take their turn and continue the game | 2 | 2 |
| 5 | A simple game is over | As a player, I want to know when a simple game is over, so I know who won the game, so I can stop playing or start a new game | 3 | 2 |
| 6 | Make a move in a general game | As a player, I want to make a move in a general game so the other player can take their turn and continue the game | 2 | 3 |
| 7 | A general game is over | As a player, I want to know when a general game is over, so I know who won the game, so I can stop playing or start a new game | 3 | 3 |

**Acceptance Criteria**

| **User Story ID and Name** | **AC ID** | **Description of Acceptance Criteria** | **Status(completed, to do, To Do)** |
| --- | --- | --- | --- |
| 1.Choose a board size | 1.1.1 | **Player successfully chooses a board size**  **Given:** game window with game options  **When:** player enters a valid board size  **Then:** wait for player to click “Start” button | To Do |
| 1.1.2 | **Player fails to enter a number equal to or greater than the minimum allowable board size**  **Given:** game window with game options  **When:** player enters a board size less than the minimum allowable size  **Then:** set board size to minimum allowable size | To Do |
| 1.1.3 | **Player fails to enter an integer board size**  **Given:** game window with game options  **When:** player enters a non-integer board size  **Then:** set board size to the floor of entered number  **And:** wait for player to click “Start” button | To Do |
| 1.1.4 | **Player fails to enter a board size equal to or less than max board size allowed**  **Given:** game window with game options  **When:** player enters a board size greater than max allowable size  **Then:** set board size to max size allowed  **And:** wait for player to click “Start” button | To Do |
| 1.1.5 | **Player fails to enter a numerical board size**  **Given:** game window with game options  **When:** player enters a non-numerical board size  **Then:** set board size to minimum allowable size  **And:** wait for player to click “Start” button | To Do |
| 2.Choose the game mode of a chosen board | 2.1.1 | **Player successfully chooses simple game mode**  **Given:** game window with game options  **When:** player selects simple game mode option  **Then:** wait for player to click “Start” button | To Do |
| 2.1.2 | **Player successfully chooses general game mode**  **Given:** game window with game options  **When:** player selects general game mode option  **Then:** wait for player to click “Start” button | To Do |
| 3.Start a new game of the chosen board size and game mode | 3.1.0 | **Player successfully starts a game with the selected game mode and chosen board size**  **Given:** board size, and game mode  **When:** player clicks “Start” button  **Then:** set up game for selected game mode  **And:** create board with chosen board size  **And:** begin with random player turn | To Do |
| 4.Make a move in a simple game | 4.1.1 | **Player 1 successfully makes a valid move**  **Given:** ongoing game  **And:** Player 1’s turn  **When:** player selects a valid cell  **Then:** update the cell in the game board  **And:** switch turn to Player 2 | To Do |
| 4.1.2 | **Player 2 successfully makes a valid move**  **Given:** ongoing game  **And:** Player 2’s turn  **When:** player selects a valid cell  **Then:** update the cell in the game board  **And:** switch turn to Player 1 | To Do |
| 4.1.3 | **Player makes an invalid move(Occupied Cell)**  **Given:** ongoing game  **When:** player selects occupied cell  **Then:** ignore move  **And:** wait for player to select valid, unoccupied cell | To Do |
| 4.1.4 | **Player makes an invalid move(Invalid Index)**  **Given:** ongoing game  **When:** player selects invalid index  **Then:** ignore move  **And:** wait for player to select valid, unoccupied cell | To Do |
| 5.A simple game is over | 5.1.1 | **Player 1 successfully completes an “SOS”**  **Given:** ongoing game  **And:** Player 1’s turn  **When:** player makes valid move  **And:** player move completes an “SOS”  **Then:** indicate the “SOS” was made by Player 1  **And:** end game  **And:** display Player 1 wins | To Do |
| 5.1.2 | **Player 2 successfully completes an “SOS”**  **Given:** ongoing game  **And:** Player 2’s turn  **When:** player makes valid move  **And:** player move completes an “SOS”  **Then:** indicate the “SOS” was made by Player 2  **And:** end game  **And:** display Player 2 wins | To Do |
| 5.1.3 | **Player makes a non-winning move that occupies last cell**  **Given:** ongoing game  **And:** only 1 cell unoccupied  **When:** player makes a valid move  **Then:** end game  **And:** display the game is a draw | To Do |
| 6.Make a move in a general game | 6.1.1 | **Player 1 successfully makes a valid move**  **Given:** ongoing game  **And:** Player 1’s turn  **When:** player selects a valid cell  **Then:** update the cell in the game board  **And:** switch turn to Player 2 | To Do |
| 6.1.2 | **Player 2 successfully makes a valid move**  **Given:** ongoing game  **And:** Player 2’s turn  **When:** player selects a valid cell  **Then:** update the cell in the game board  **And:** switch turn to Player 1 | To Do |
| 6.1.3 | **Player 1 successfully completes an “SOS”**  **Given:** ongoing game  **And:** Player 1’s turn  **When:** player makes a valid move  **And:** player move completes an “SOS”  **Then:** indicate the “SOS” was created by Player 1  **And:** increment Player 1’s score by 1  **And:** switch to Player 2’s turn | To Do |
| 6.1.4 | **Player 2 successfully completes an “SOS”**  **Given:** ongoing game  **And:** Player 2’s turn  **When:** player makes a valid move  **And:** player move completes an “SOS”  **Then:** indicate the “SOS” was created by Player 2  **And:** increment Player 2’s score by 1  **And:** switch to Player 1’s turn | To Do |
| 6.1.5 | **Player makes an invalid move(Invalid Index)**  **Given:** ongoing game  **When:** player selects invalid index  **Then:** ignore move  **And:** wait for player to select valid, unoccupied cell | To Do |
| 7.A general game is over | 7.1.1 | **Player move occupies last cell(Winner)**  **Given:** ongoing game  **And:** only 1 cell unoccupied  **When:** player makes a valid move  **And:** player move occupies last cell  **And:** players do not have same score  **Then:** end game  **And:** display player with higher score is the winner | To Do |
| 7.1.2 | **Player move occupies last cell(Draw)**  **Given:** ongoing game  **And:** only 1 cell unoccupied  **When:** player makes a valid move  **And:** player move occupies last cell  **And:** players have same score  **Then:** end game  **And:** display the game is a draw | To Do |