

Reference Story Number	Story Pool	Story Source	Summary
1	1	EECS 581 Project 1	Create two 10x10 grids (one for player 1 and one for player 2) with labeled columns (A-J) and rows (1-10). Each tile should have variables to track its status, such as whether it is empty, occupied by a ship, or has been hit, including details about the specific ship.
2	1	EECS 581 Project 1	The game concludes when all of the opponent's ships are sunk, requiring a condition check at the end of each player's turn to confirm if all ships are destroyed and the game is over.
3	1	EECS 581 Project 2	Develop an easy AI for battleship that randomly selects tiles to fire upon, requiring no complex logic for ship targeting or tactical decision-making beyond random tile selection.
4	2	EECS 581 Project 1	Implement a system where a ship is marked as destroyed once all of its spaces are hit. This will trigger a visual update on the board to indicate that the ship has been sunk.
5	2	EECS 581 Project 2	Develop a function to enable the AI to automatically place player 2's battleships on the grid for single-player mode, ensuring the placements are randomized and follow valid ship placement rules.
6	2	EECS 581 Project 1	Implement a win condition that triggers when all of the opponent's ships are sunk. The system should then end the game and display a message declaring the player as the winner.
7	3	EECS 581 Project 2	Develop a hard AI that always knows the locations of the player's ships and successfully lands a hit on one of the ships in every turn, ensuring a challenging opponent.
8	3	EECS 581 Project 1	Develop a system where players can select the number of ships to use (ranging from 1 to 5). The system should track each ship's size, ranging from 1x1 to 1x5, and display which ships are selected, depending on the total number chosen for the game.
9	3	EECS 581 Project 2	Develop a system that allows players to choose between one or two-player modes. If a one-player mode is selected, the system should provide AI difficulty options (easy, medium, hard) and modify the game loop accordingly to handle AI turns and determine if the AI has won.
10	5	EECS 581 Project 1	Implement a turn-based system where players alternate in firing at each other's boards. The system must give immediate feedback indicating whether the shot was a hit or miss, ensure coordinate checks for accuracy, and keep each player's grid hidden from the opponent.
11	5	EECS 581 Project 2	Develop a medium-difficulty AI for the game. This AI should fire randomly until it hits a ship, then continue firing in adjacent spaces to identify further hits and sink the entire ship.
12	5	EECS 581 Project 2	Develop a leaderboard system that saves player scores to a file. The system should also allow for loading these scores for viewing across multiple game sessions for long-term score tracking.
13	8	EECS 581 Project 2	Implement a scoreboard that tracks and displays player scores, awarding 100 points per hit and deducting 1 point per miss. Integrate the scoreboard with the existing leaderboard, allowing game winners to enter their names and store their scores across game sessions, ensuring persistent data for long-term player rankings.
14	8	EECS 581 Project 2	Ensure that all required functionalities from Project 1 are fully functional, even if inherited code from previous teams is incomplete. This involves learning how existing features work to properly implement updates and bug fixes for the game.
15	8	EECS 581 Project 2	Develop comprehensive documentation outlining the programming language, IDE, and any other relevant tools used. This should include details needed for both playing the game and for maintaining or updating the game's code for future development.
16	13	EECS 581 Project 1	Implement a dynamic player view that displays their board and ships, and provides real-time feedback on hits and misses. The visual design should update interactively based on player actions, reflecting the state of the game as it progresses.
17	13	EECS 581 Project 1	Develop a system allowing players to place and orient their ships on the grid in secret. This system should support ship rotation, prevent overlapping ships, and ensure that ships cannot extend beyond the grid's boundaries, providing clear and valid ship placement for both players.
18	13	EECS 581 Project 2	Refactor and debug the code to ensure that different components of the program integrate smoothly. This process involves identifying and fixing inconsistencies, ensuring proper communication between various modules of the game.