Programming Fundamentals Project

Battleship Simulator – The Game

CLO 3 – Deadline: November 30, 2025

1. Instructions

1. Make sure that you read and understand each instruction. If you have any questions or comments, you are encouraged to discuss your problems with your colleagues or instructors on Google Classroom.
2. If there is a syntax error in the code, zero marks will be awarded for that part of the project.
3. Keep a backup of your work to prevent data loss and avoid last-minute submissions.
4. Displayed output should be well presented and neatly formatted. Use appropriate comments and indentation in your source code.
5. Please ensure that only concepts covered in class are used for the project. File handling is a self-learning concept that you will use in this project.
6. Work collaboratively and ensure both team members contribute equally.
7. Only submit your project .cpp file. The file name should follow the format: Student1_RollNo-Student2_RollNo. Only one group member should submit the file.
8. Each student is responsible for ensuring the final file is correct, virus-free, and accessible. Corrupted or inaccessible files will result in zero marks.
9. Start early so that you can finish on time!

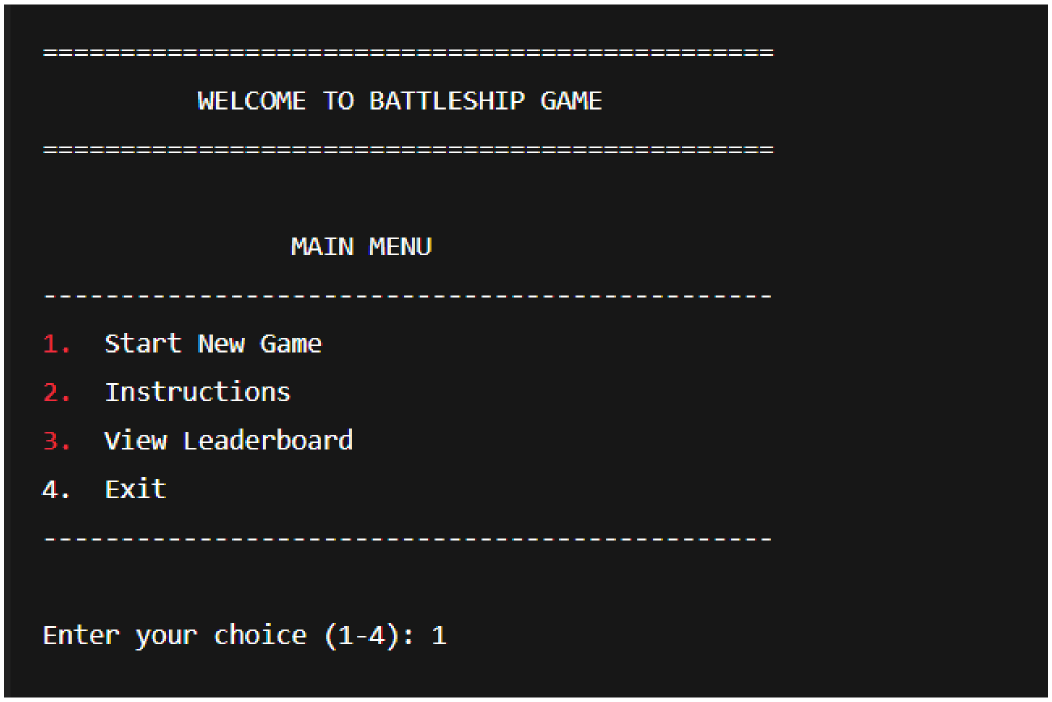
2. Game Overview

Battleship is a two-player strategy game (versus Human or Computer). Each player tries to sink the opponent's fleet by guessing the positions of their ships. Strategy and luck both play a part in winning. You will implement the Battleship game in C++ with interactive menus, scoring, and file-based leaderboard management.

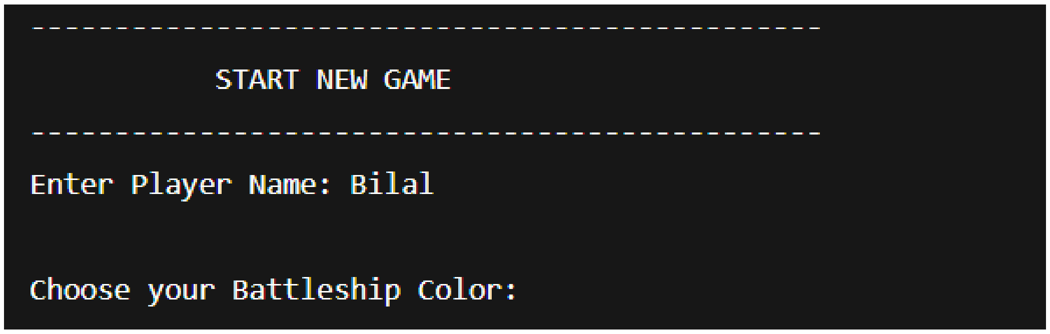
3. Game Start Menu

- Start New Game – Enter player's name and select battleship color.
- Instructions – Display the rules of the game.
- View Leaderboard – Show top 10 scores from highscores.txt (if available). If the current score is among the top 10, update the file accordingly.
- Exit – Quit the game.

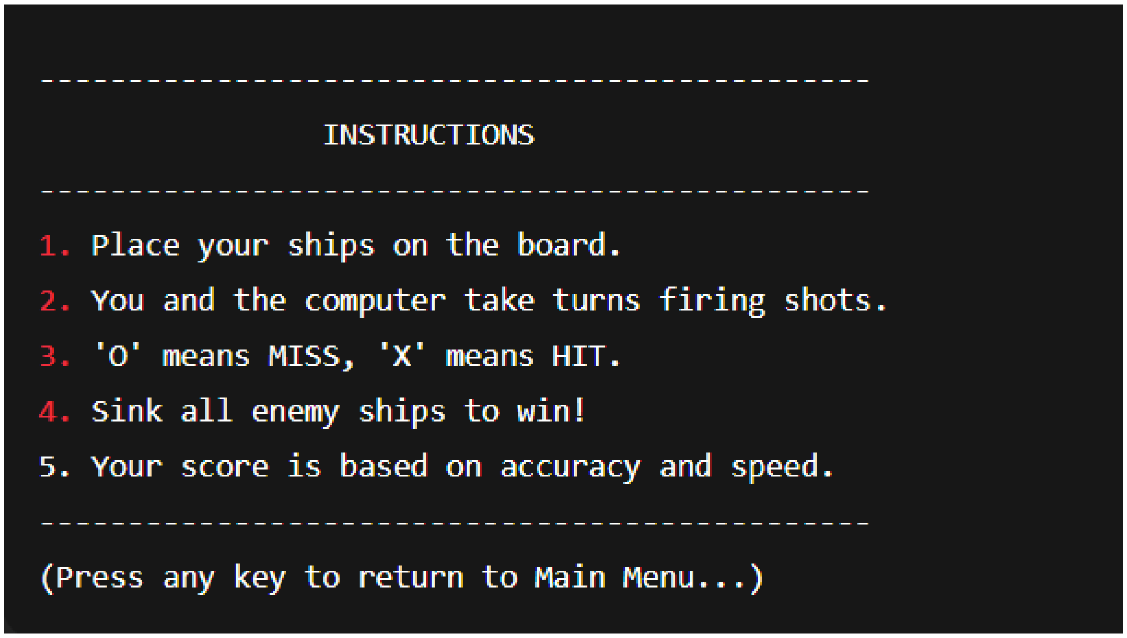
Sample output screens for above menu is as follows:



```
==============================================
          WELCOME TO BATTLESHIP GAME
==============================================

                MAIN MENU
----------------------------------------------
1.  Start New Game
2.  Instructions
3.  View Leaderboard
4.  Exit
----------------------------------------------

Enter your choice (1-4): 1
```



```
------------------------------------------------
           START NEW GAME
------------------------------------------------
Enter Player Name: Bilal

Choose your Battleship Color:
```



```
---------------------------------------------
              INSTRUCTIONS
---------------------------------------------
1. Place your ships on the board.
2. You and the computer take turns firing shots.
3. 'O' means MISS, 'X' means HIT.
4. Sink all enemy ships to win!
5. Your score is based on accuracy and speed.
---------------------------------------------
(Press any key to return to Main Menu...)
```

4. The Board

When the user starts a new game, they can choose to play against another player or the computer.

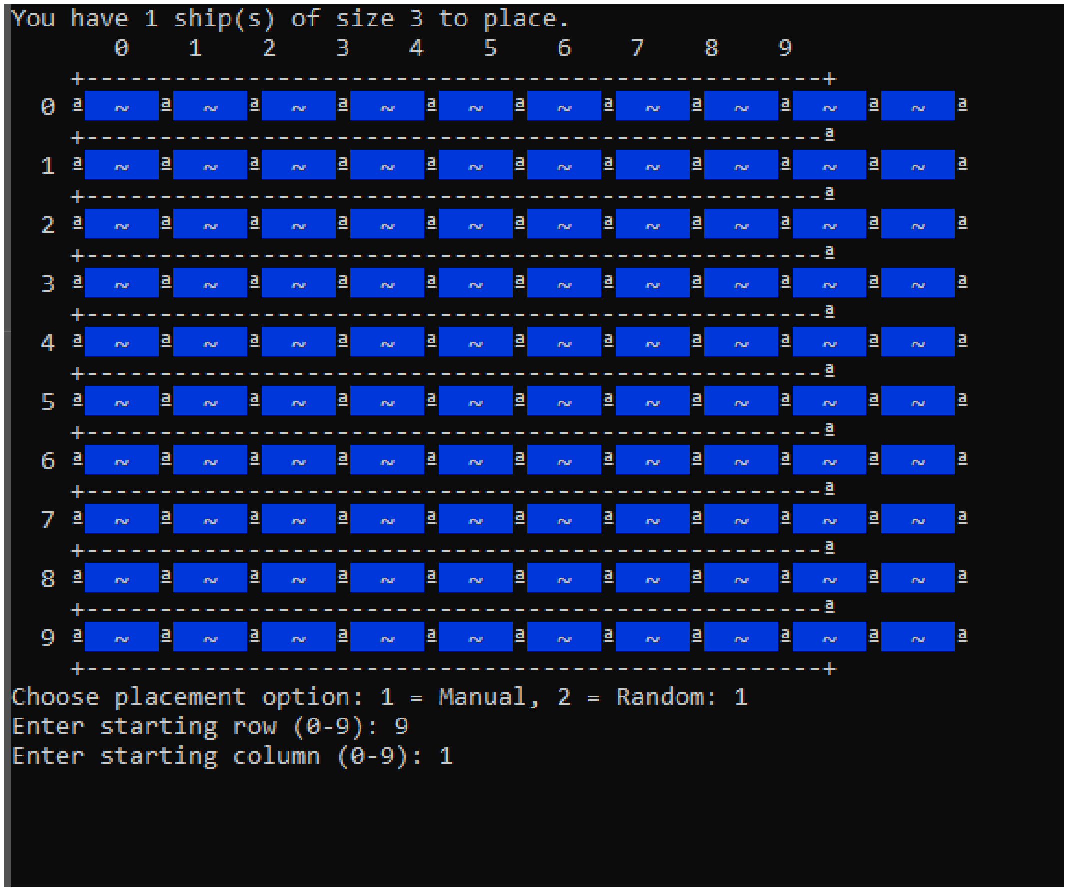
Draw a 10x10 battle grid along with a sidebar showing available ships. Ships are placed horizontally by default, but can be rotated vertically. Ensure ships do not overlap or exceed the grid boundaries. A 'Random' option should also be available for automatic placement.

Ship Types and Sizes:

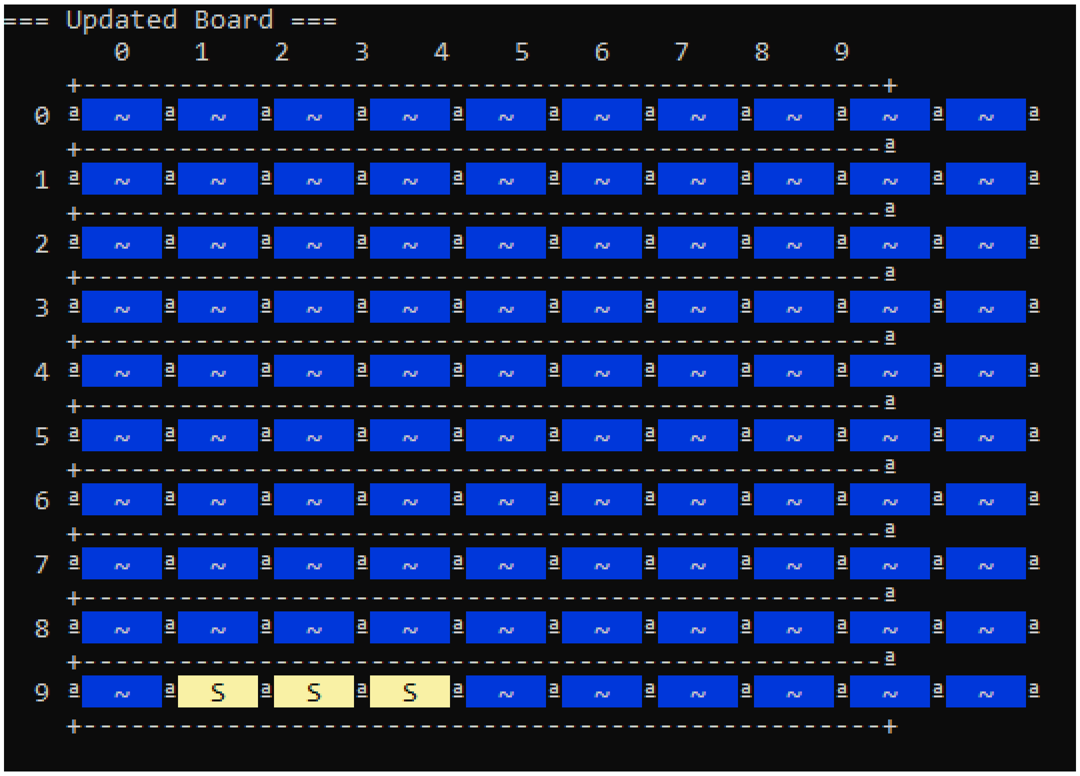
- Aircraft Carrier – 5 blocks
- Battleship – 4 blocks
- Destroyer – 3 blocks
- Submarine – 3 blocks
- Patrol Boat – 2 blocks

You will show the length of a battleship with number of cells it occupies on the grid. When user selects a ship, the user is required to provide row and column of the gird. By default, the ship will be placed horizontally. If user select row 9 and column 1 for the ship size 3, the ship will be place on 9-1,9-2,9-3 (as shown in picture). The player can rotate it vertically. Make sure that boundaries are not violated. E.g. if user select Row 1 and Col 9 for ship size 3, it cannot be placed horizontally. You are required to do input validation and display appropriate message.

Example of ship placement is as follows



```
You have 1 ship(s) of size 3 to place.
       0    1    2    3    4    5    6    7    8    9
    +---------------------------------------------------+
  0 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  1 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  2 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  3 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  4 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  5 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  6 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  7 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  8 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  9 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------+
Choose placement option: 1 = Manual, 2 = Random: 1
Enter starting row (0-9): 9
Enter starting column (0-9): 1
```

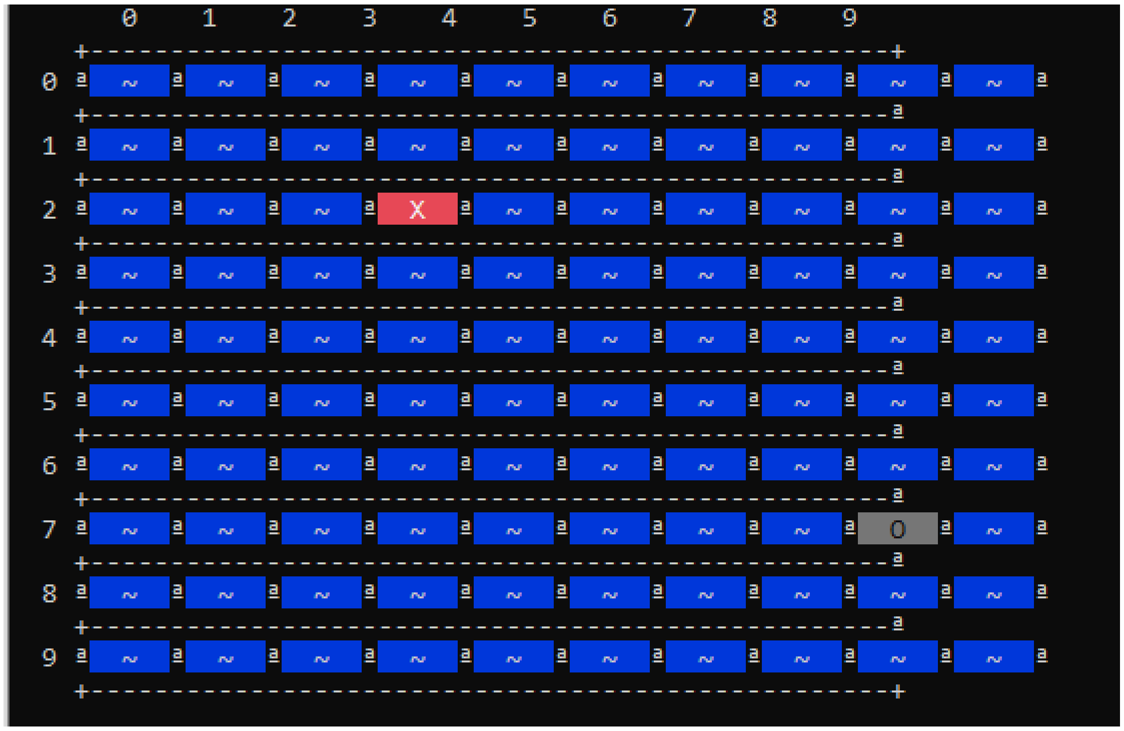


```
=== Updated Board ===
       0    1    2    3    4    5    6    7    8    9
    +---------------------------------------------------+
  0 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  1 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  2 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  3 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  4 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  5 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  6 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  7 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  8 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  9 ª  ~  ª  S  ª  S  ª  S  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------+
```

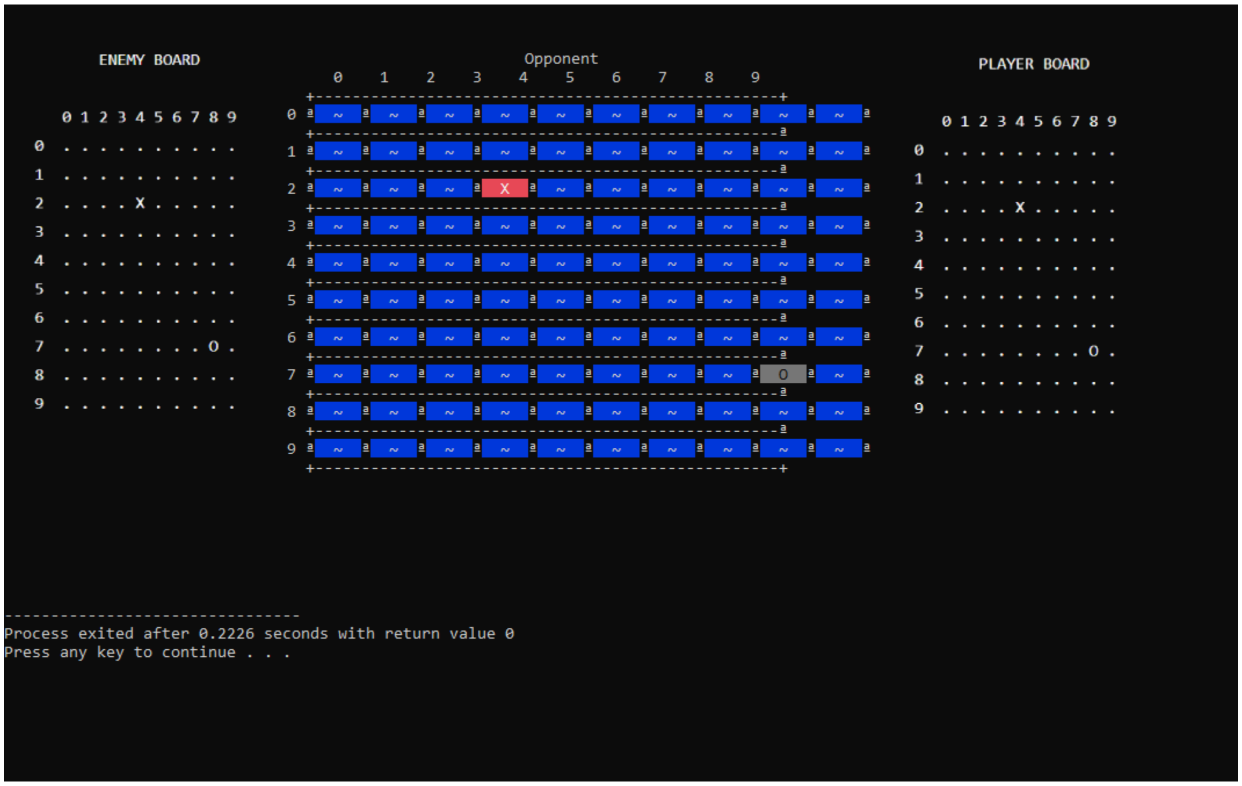
5. Game Play

- Each player (or computer) places 5 ships on their grid.
- Players alternate turns with number of ships + 1 shots until one player’s ships are completely destroyed.
- On each turn, the player enters row and column coordinates to attack.
- Cells already attacked (marked X or O) cannot be selected again.
- A hit is marked with red 'X', and a miss with grey 'O'.
- Each hit gives +10 points; each miss deducts 1 point.
- When all cells of a ship are hit, color them pink to indicate the ship is sunk.
- The player who destroys all enemy ships first wins.

A sample board is attached below.



```
        0     1     2     3     4     5     6     7     8     9

    +---------------------------------------------------+
  0 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  1 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  2 ª  ~  ª  ~  ª  ~  ª  X  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  3 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  4 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  5 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  6 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  7 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  O  ª  ~  ª
    +---------------------------------------------------ª
  8 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------ª
  9 ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª  ~  ª
    +---------------------------------------------------+
```



```
        ENEMY BOARD                              Opponent                                        PLAYER BOARD
                               0    1    2    3    4    5    6    7    8    9
                            +--------------------------------------------------+
    0 1 2 3 4 5 6 7 8 9   0 ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª        0 1 2 3 4 5 6 7 8 9
                            +--------------------------------------------------ª
 0  . . . . . . . . . .   1 ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª     0  . . . . . . . . . .
 1  . . . . . . . . . .     +--------------------------------------------------ª     1  . . . . . . . . . .
 2  . . . . X . . . . .   2 ª ~  ª ~  ª ~  ª X  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª     2  . . . . X . . . . .
 3  . . . . . . . . . .     +--------------------------------------------------ª     3  . . . . . . . . . .
 4  . . . . . . . . . .   3 ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª     4  . . . . . . . . . .
 5  . . . . . . . . . .     +--------------------------------------------------ª     5  . . . . . . . . . .
 6  . . . . . . . . . .   4 ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª     6  . . . . . . . . . .
 7  . . . . . . . . O .     +--------------------------------------------------ª     7  . . . . . . . . O .
 8  . . . . . . . . . .   5 ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª     8  . . . . . . . . . .
 9  . . . . . . . . . .     +--------------------------------------------------ª     9  . . . . . . . . . .
                          6 ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª
                            +--------------------------------------------------ª
                          7 ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª O  ª ~  ª
                            +--------------------------------------------------ª
                          8 ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª
                            +--------------------------------------------------ª
                          9 ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª ~  ª
                            +--------------------------------------------------+

--------------------------------
Process exited after 0.2226 seconds with return value 0
Press any key to continue . . .
```

6. Function Prototypes

- // Initialize the board with '~' in every cell
  void initializeBoard(char board[10][10]);
- // Display the board (optionally hide ships)
  void printBoard(char board[10][10], bool hideShips);
- // Clear the console screen
  void clearScreen();
- // Randomly places all ships (used for computer or player 2)
  void setRandomShips(char board[10][10]);
- // Handles Player vs Computer mode
  void vsComputer(char playerBoard[10][10], char computerBoard[10][10]);
- // Handles Player vs Player mode
  void vsPlayer(char board1[10][10], char board2[10][10]);
- // Executes player attack; invalid if cell already attacked
  void playerAttack(char opponentBoard[10][10], int &playerScore);
- // Computer attacks a random valid cell
  void computerAttack(char playerBoard[10][10], int &computerScore);
- // Updates player score after each hit
  void updateScore(int &score);
- // Displays main game menu and lets user select mode
  void showGameMenu();

- // Compares final scores and displays the winner or draw
  void displayResult(int player1Score, int player2Score);
- // Clears all cells on the board to '~'
  void clearBoard(char board[10][10]);

Work Division

Common features

// Initialize the board with '~' in every cell

void initializeBoard(char board[10][10]);

// Display the board (hiding ships)

void printBoard(char board[10][10], bool hideShips);

// Clear the console screen between turns

void clearScreen();

// Show Leaderboard

void showLeaderboard(const string playerNames[], const int scores[], int totalPlayers);

Student 1 Responsibilities	**Student 2 Responsibilities**
Ship placement for computer (random)	Ship placement for player (for vs player)
Vs Computer Gameplay	Player vs Player mode
Computer attack logic (random hit)	Player attack logic (valid attack only once per cell)
Clear board function	Game menu and mode selection
Scoring system (counting hits)	Win/Loss detection

Function Prototypes
// Randomly places all ships (used for computer or player 2)
void setRandomShips(char board[10][10]);
 	
// Handles full Player vs Computer mode
void vsComputer(char playerBoard[10][10], char computerBoard[10][10]);
 	
// Computer attacks a random valid cell
void computerAttack(char playerBoard[10][10], int &computerScore);
	Function Prototypes
	// Allows player to manually set ships (horizontal or vertical)
	void setRandomShips(char board[10][10]);
	
	// Handles full Player vs Player mode
	void vsPlayer(char board1[10][10], char board2[10][10]);
	
	// Executes player attack; invalid if cell already attacked
	void playerAttack(char opponentBoard[10][10], int &playerScore);

// Clears all cells on the board to '~' void clearBoard(char board[10][10]);	// Displays main game menu and lets user select mode void showGameMenu();
// Updates player score after each successful hit void updateScore(int &score);	// Compares final scores and displays the winner or draw void displayResult(int player1Score, int player2Score);

7. Validation Rules

- Ships must not overlap existing ships.
- Ships must fit entirely on the board.
- Ships can be placed only horizontally or vertically.
- Players cannot attack the same or invalid cells twice.
- Winner is determined by total score or destruction of all ships.

Happy Coding 😊