# Introduction to Programming
# 2nd Project

Programming Group
guido.salvaneschi@unisg.ch

University of St. Gallen – December 5th, 2024

## Declaration of Authorship

"I hereby declare

- that I have solved this project without any help from others and without the use of documents and aids other than explicitly stated;
- that I have mentioned all the sources used and that I have cited them correctly according to established academic citation rules;
- that I have acquired any immaterial rights to materials I may have used such as images or graphs, or that I have produced such materials myself;
- that I will not pass on copies of this work to third parties or publish them without the University's written consent;
- that I am aware that my work can be electronically checked for plagiarism and that I hereby grant the University of St. Gallen copyright in accordance with the Examination Regulations in so far as this is required for administrative action;
- that I am aware that the University will prosecute any infringement of this declaration of authorship and, in particular, the employment of a ghostwriter, and that any such infringement may result in disciplinary and criminal consequences which may result in my expulsion from the University or my being stripped of my degree"

*By submitting a solution to this project, you confirm through conclusive action that you are submitting the "Declaration of Authorship," that you have read and understood it, and that it is true.*

**Warning:** You must work individually and without the help of any other person on this project. All submissions will be checked for plagiarism. If we suspect plagiarism or if it appears that the project was completed with significant AI assistance, we may ask you for a clarification meeting and a personal code presentation with potential point deductions. Found plagiarism leads to failing the entire exam.

Please follow HSG regulations on written work, including DDoS Section 5 `https://erlasse.unisg.ch/lexoverview-home/lex-II_B_1_20`.

Submission deadline: December 19, 2024, 16:00.

Task 1:

Implement the program specified below in a module file `battleships.py` and submit it before the deadline to codePost. Make sure your program, i.e., its `main()` function, is only executed when the module is directly executed, not when it is imported in another Python program. Only submit your `battleships.py` file. The provided `ui.py`, `test_battleships.py`, `conftest.py` and `pytest.ini` files are available as provided on the submission platform and cannot be overwritten.

Make sure your code is of high quality, i.e., it uses Python adequately, is well-structured and formatted, uses self-explaining identifier names, and contains reasonable comments that concisely document all non-trivial design decisions. According to the published grading scheme, your implementation will be assessed for functionality (50%), compliance (20%), and style (30%).

**Info:** We provide some tests in `test_battleships.py`. You can use them to test your implementation locally. To do so, download the files `test_battleships.py`, `conftest.py`, and `pytest.ini` to the directory with your `battleships.py` and run `pytest` in that directory. The exactly same tests will be available and executed on the submission platform every time you submit. Make sure, these tests work also on your submission on codePost to ensure compatibility. You can update your submission until the deadline. Only the final submission will be considered for grading.

Note that such tests only cover *basic* functionalities for your early feedback and are not indicative of the fact that the application fully implements the functionalities required for the project.

To grade your submission, we will further verify that the application implements all the requirements presented in this document.

**Warning:** Submissions that are not compatible with the submission platform are treated as missing submissions. If you should face technical issues with the submission platform, contact **both** TAs as early as possible (send an email to david.spielmann@unisg.ch **and** kai.schultz@student.unisg.ch). The TAs are not obliged to respond shortly before the deadline nor to fix non-systemic technical issues, i.e., issues that are not caused by the platform or its configuration.

**Info:** We do not prioritize the performance of your implementation. However, if your implementation is very inefficient and you upload your submission to CodePost, the tests may time out. In such cases, you will see failed tests with the message: `Operation Timed Out. No output received.` If this occurs, we recommend testing your implementation locally. When grading your submission, if we observe that tests run into a timeout, we will rerun the tests. If they pass upon rerunning, you will receive the corresponding points.

## Battleship Game

The game of Battleship is a two-player game where each player has a grid on which they place their ships. The grid is a square grid of size eight by eight. In a first step, each player places five ships on their grid. Afterwards, the players take turns to shoot at the opponent's grid. The goal is to sink all of the opponent's ships. The game ends when all ships of one player are sunk.


Figure 1: Battleship game illustration.[1]

## Step-by-step Program Description

1. The game starts in the menu (see "Menu").
   (i) The menu header is displayed (text above the menu), announcing the name of the game. All menu items are displayed to the users.
   (ii) The users are prompted to select a menu item by entering its number. On invalid input, the prompt is repeated until a valid item number is provided.
   (iii) The users have the following choices:
   - "Play Battleships": This is the "Game of Battleships". Two players first place their ships on a grid of size eight by eight, and then take turns to shoot at the opponent's grid. The primary goal is to be the first player to sink all of the opponent's ships. The program continues with step 2.
   - "Scoreboard": The program continues with step 6.
   - "Exit": The program ends.
2. The users are asked to enter their names for the next game (see "Entering Names").
   (i) The enter header is displayed (text above the input prompt), announcing that the users have to enter their names.
   (ii) The users are prompted to enter the name for player A for the next game. The name must have at least one character. On invalid input, the prompt is repeated until a valid name is provided.
   (iii) The users are prompted to enter the name for player B for the next game. The name must have at least one character and be different from the name of player A. On invalid input, the prompt is repeated until a valid name is provided.
   (iv) The program continues with step 3 with an empty grid and player A.
3. The users are asked to place their ships on the grid (see "Place Ships"). Each player has their own grid.
   (i) The place header is displayed (text above the grid), announcing the name of the player who has to place the ships, the type of ship, and the length of the ship, followed by the grid.
   (ii) Player A is prompted to enter the starting square and the ending square of the ship. The input is valid only if the ship is either placed horizontally or vertically and does not overlap with other ships. The input is the row and column of the starting square, separated by one or more whitespace characters, followed by a comma. Next, the row and column of the ending square, separated by one or more whitespace characters. If the input is invalid, the prompt is repeated until a valid input is provided. Player A is asked to place the ships on the grid in the following order: speedboat of length 2, destroyer of length 4, attacker of length 3, a second attacker of length 3, and an aircraft carrier of length 5.

[1] Image taken from `https://www.toysrus.ca/en/Battleship-Classic-Board-Game--Strategy-Game/04997A77.html`. Accessed on November 22, 2024.

(iii) Player B is asked to place the same ships on the grid in the same order. On invalid input, the prompt is repeated until a valid input is provided.

(iv) The program continues with step 4.

4. A turn is played (see "Game Round").

   (i) The turn header (text above the grid in every turn, see "Game Round") is displayed, announcing the name of the player whose turn it is.

   (ii) The current grid is displayed.

   (iii) The current player is prompted to enter the number of the row and the number of the column where they wish to shoot, separated by one or more whitespace characters. Both row and column numbers are positive, i.e., 1 or higher. The input is valid only if the row and column numbers are within the grid and the corresponding square has not been shot at before. If the input is invalid, the prompt is repeated until a valid row-column pair is provided.

   (iv) The shot is added to the grid. If the shot hits a ship, this is indicated on the grid with an "X". If the shot misses, this is indicated with an "0".

   (v) If a user won, i.e., all ships of the opponent are sunk, the program continues with step 5. Otherwise, the program continues with step 4 and with the other player.

5. The game result is shown (see "Game Result").

   (i) If all ships of one player are sunk, a message is displayed that the game is over.

   (ii) The final grid is displayed.

   (iii) The name of the winner is displayed.

   (iv) The winner's score in the scoreboard is increased by 1.

   (v) The users are prompted to press enter to return to the menu. On any input, the program continues with step 1.

6. The scoreboard is shown (see "Scoreboard").

   (i) The scoreboard, a list of all player names with their score, is displayed to the users.

   (ii) The users are prompted to press enter to return to the menu. On any input, the program continues with step 1.

## Implementation Scaffold

Your implementation of the `battleships.py` module must implement the following functions. Other modules have to be able to import them from your module file and use them as described below. You may add additional functions to your module to structure your code and all functions below definitely may call other functions to implement their functionality. As reference or starting point, you can use the provided scaffold file, which defines all these functions as required without an implementation.

1. `main()` is the entry point into the game and runs its full implementation. Returns `None`.
2. `menu()` displays the menu to the user and repeatedly asks for the user's choice until a valid answer was provided. Returns the number of the selected menu item as integer.
3. `save_scoreboard(scoreboard)` saves the scoreboard provided in the parameter `scoreboard` to the file `scoreboard.dat`. If the file already exists, it is overwritten. Returns `None`.
4. `load_scoreboard()` loads the scoreboard from the file `scoreboard.dat`. Returns the scoreboard without entries where the score is 0. If the file does not exist, is not readable, or does not contain data in the expected format, an empty scoreboard (an empty dictionary) is returned.

5. `is_game_won(grid)` receives the current grid state in parameter `grid`. The function evaluates whether a player won. In particular, the game is won if a player has sunk all ships of the opponent. If a player won, the function returns `True`, otherwise `False`.

6. `play_turn(grid_a, grid_b, player_name, is_player_a)` receives the current grids of player A and player B in parameters `grid_a` and `grid_b`, respectively. The name of the player whose turn it is as string in `player_name` and whether the player is player A as boolean in `is_player_a`. The function first displays a turn start message with the player's name whose turn it is and which board the player has to attack. Then it displays the two grids and prompts the user to enter their chosen row and column. The prompt is repeated until a valid input is provided. Finally, the function returns a boolean value, `True` if the player hit a ship, `False` if the player missed.

7. `is_ship_position_possible(length, start_row, start_col, end_row, end_col, grid)` receives the length of the ship in parameter `length` as an integer, the row and column of the starting square in parameters `start_row` and `start_col`, the row and column of the ending square in parameters `end_row` and `end_col`, and the current grid in parameter `grid`. The function evaluates whether the ship can be placed on the grid. Returns `True` if the ship can be placed, `False` otherwise.

8. `position_ships(player, rows, cols, ships)` receives the name of the player as string in parameter `player`, the number of rows and columns in the grid in parameters `rows` and `cols`, and a list of tuples with the ship names and their lengths in parameter `ships`. The function first displays a header with the name of the player and the ship to be placed, followed by the length of the ship. Then it displays the grid and prompts the user to enter the starting and ending square of the ship. The prompt is repeated until a valid input is provided. Finally, the function returns the grid (i.e., a two-dimensional list) with the placed ships.

9. `play_battleships(ships=SHIPS)` receives as optional parameter `ships` a list of tuples with the ship names and their lengths. The default value is the defined constant `SHIPS`. first asks the users for their names. It then run the game until a player has won. It then shows the result of the game and prompts the user to press enter to return to the menu. On any input, the function returns the name of the player that won as string.

In all functions above, the *scoreboard* is a dictionary, where the keys are player names as strings and the values are their respective score as integer.

In all functions above, the *grid* is a two-dimensional list. The length of the outer list is the number of columns in the game's grid. The outer list represents the columns from left to right. Each inner list has the length of the number of rows in the game's grid. Each value in the inner lists defines the state of a square in the grid. Each player has their own grid.

Values are `None` if no ship is present and was not shot at, `True` if a ship is present and was not shot at, `False` if a ship was hit, `'miss'` if a shot missed a ship. For example, the following indices are valid `grid[0..7][0..7]`. If `gridA[0][0]` is `None`, playerA did not place a ship at the square in the first column and the first row. If `gridA[3][2]` is `True`, player A has placed a ship in the square in the fourth column from the left and the third row from the top. If `gridB[1][5]` is `False`, player A has shot at the square in the second column from the left and sixth row from the top and hit a ship of player B. If `gridB[6][4]` is `'miss'`, player A has shot at the square in the seventh column from the left and fifth row from the top and missed a ship of player B.

## UI Module

The provided `ui.py` module implements the user interaction of the game. It implements the following functions, which must be used in your Battleship implementation for all user interaction. Besides the documentation below, looking at the provided code and comments may clarify what each function does. The representation of scoreboard and grid is the same as described above in "Implementation Scaffold."

**Warning:** Use `ui.py` as provided and do not change it. On the submission platform, only the provided version will be available. You cannot replace it with your own, altered implementation.

- `display_grid(grid)` displays the grid as provided in the parameter `grid` Returns `None`.
- `display_game(gridA, gridB)` displays the game, where the current grids of both players are provided in `gridA` and `gridB`. Returns `None`.
- `display_headline(headline)` displays a headline in uppercase that is provided as string in `headline`. Returns `None`.
- `display_menu(items)` displays the menu screen. It receives the menu items as list of strings in `items` and displays the items in the order of occurrence in the list as enumeration, starting with the number 1. Returns `None`.
- `display_message(message)` displays a message that is provided as string in `message`. Returns `None`.
- `display_scoreboard(scoreboard)` displays the scoreboard received in the `scoreboard`. Returns `None`.
- `display_turn_start(player_name, is_player_a)` receives the name of the player as string in `player_name` and whether the player is player A as boolean in `is_player_a`. The function displays the start of a new turn, showing the player's name and the board they are attacking. Returns `None`.
- `prompt(message)` displays a prompt with the string provided in `message` and waits for a line of user input on STDIN. Returns the string read from STDIN without trailing `\n` linefeed character.

To clear the screen, `display_headline` and `display_turn_start` print a number of empty lines at the beginning.

## Implementation Constraints

The program has to adhere to the following constraints:

1. The program does only use the provided UI module for input and output, i.e., it does not print to STDOUT or read from STDIN (e.g., the program does not use `print` and `input`).
2. In one execution, the program reads at most one file at most once.
3. The program persists scoreboard updates as soon as possible such that no data is lost if the program is interrupted directly after the respective game function in the scaffold completed.
4. No other import than `import ui` and `import pickle` is used.
5. The game logic (asking for the names, deciding whether the game continues, deciding whose turn it is, performing a turn, evaluating whether a player won, showing the result of the game, and the flow between these actions) is implemented at most once.

## User Interactions

In the following, you find examples of interactions with the users in the "Step-by-step Program Description." They can and shall be implemented by solely using the provided "UI Module." Make sure that your implementation exactly resembles the interactions below.

### Menu

Command Line

```
MENU BATTLESHIPS

1. Play Battleships
2. Scoreboard
3. Exit

Enter the number of your choice:
```

An interaction with four invalid inputs followed by the selection of the exit looks like:

Command Line

```
MENU BATTLESHIPS

1. Play Battleships
2. Scoreboard
3. Exit

Enter the number of your choice: myvalue
Enter the number of your choice: 0
Enter the number of your choice: -1
Enter the number of your choice: 7
Enter the number of your choice: 3
```

### Scoreboard

Command Line

```
SCOREBOARD BATTLESHIPS

1. Chris (83)
2. Peter (53)
3. Stewie (48)
4. Lois (35)
5. Meg (20)
6. Brian (10)

Press ENTER to return to the menu:
```

An empty scoreboard is represented by an empty dictionary results in:

Command Line

```
SCOREBOARD BATTLESHIPS

no scores available

Press ENTER to return to the menu:
```

## Entering Names

An interaction configuring the player names Stewie and Brian looks like:

Command Line

```
ENTER PLAYER NAMES

Enter the name of player A: Stewie
Enter the name of player B: Brian
```

Prompts are repeated until a valid input is provided. An interaction with invalid values eventually configuring Stewie and Brian looks like:

Command Line

```
ENTER PLAYER NAMES

Enter the name of player A:
Enter the name of player A: Stewie
Enter the name of player B: Stewie
Enter the name of player B:
Enter the name of player B: Brian
```

## Place Ships

An interaction configuring the ships for player Stewie looks like:

Command Line

```
STEWIE POSITION THE SHIP SPEEDBOAT - LENGTH 2

    1   2   3   4   5   6   7   8
  +---+---+---+---+---+---+---+---+
1 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
2 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
3 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
6 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
7 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
8 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+

Select beginning and end cell: : 1 1, 1 2

STEWIE POSITION THE SHIP DESTROYER - LENGTH 4

    1   2   3   4   5   6   7   8
  +---+---+---+---+---+---+---+---+
1 | 0 | 0 |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
2 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
3 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
6 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
7 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
8 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+

Select beginning and end cell: : 3 1, 3 4
```

Command Line

```
STEWIE POSITION THE SHIP ATTACKER - LENGTH 3

    1   2   3   4   5   6   7   8
  +---+---+---+---+---+---+---+---+
1 | 0 | 0 |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
2 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
3 | 0 | 0 | 0 | 0 |   |   |   |   |
  +---+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
6 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
7 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
8 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+

Select beginning and end cell: 1 5, 3 5

STEWIE POSITION THE SHIP ATTACKER - LENGTH 3

    1   2   3   4   5   6   7   8
  +---+---+---+---+---+---+---+---+
1 | 0 | 0 |   |   | 0 |   |   |   |
  +---+---+---+---+---+---+---+---+
2 |   |   |   |   | 0 |   |   |   |
  +---+---+---+---+---+---+---+---+
3 | 0 | 0 | 0 | 0 | 0 |   |   |   |
  +---+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
6 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
7 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
8 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+

Select beginning and end cell: 7 3, 7 5
```

Command Line

```
STEWIE POSITION THE SHIP AIRCRAFT CARRIER - LENGTH 5

    1   2   3   4   5   6   7   8
  +---+---+---+---+---+---+---+---+
1 | 0 | 0 |   |   | 0 |   |   |   |
  +---+---+---+---+---+---+---+---+
2 |   |   |   |   | 0 |   |   |   |
  +---+---+---+---+---+---+---+---+
3 | 0 | 0 | 0 | 0 | 0 |   |   |   |
  +---+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
6 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
7 |   |   | 0 | 0 | 0 |   |   |   |
  +---+---+---+---+---+---+---+---+
8 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+

Select beginning and end cell: 3 8, 7 8

BRIAN POSITION THE SHIP SPEEDBOAT - LENGTH 2

    1   2   3   4   5   6   7   8
  +---+---+---+---+---+---+---+---+
1 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
2 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
3 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
6 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
7 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+
8 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+

Select beginning and end cell: 1 1, 1 2
```

## Game Round

The first rounds of a game look like:

Command Line

```
BATTLESHIPS

Stewie, it is your turn! Attack the right board.

    1   2   3   4   5   6   7   8             1   2   3   4   5   6   7   8
  +---+---+---+---+---+---+---+---+   |     +---+---+---+---+---+---+---+---+
1 |   |   |   |   |   |   |   |   |   |   1 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |     +---+---+---+---+---+---+---+---+
2 |   |   |   |   |   |   |   |   |   |   2 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |     +---+---+---+---+---+---+---+---+
3 |   |   |   |   |   |   |   |   |   |   3 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |     +---+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   |   |   |   |   4 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |     +---+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |   |   5 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |     +---+---+---+---+---+---+---+---+
6 |   |   |   |   |   |   |   |   |   |   6 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |     +---+---+---+---+---+---+---+---+
7 |   |   |   |   |   |   |   |   |   |   7 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |     +---+---+---+---+---+---+---+---+
8 |   |   |   |   |   |   |   |   |   |   8 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |     +---+---+---+---+---+---+---+---+

Please select a row and a column: 1 1

BATTLESHIPS

Brian, it is your turn! Attack the left board.

    1   2   3   4   5   6   7   8             1   2   3   4   5   6   7   8
  +---+---+---+---+---+---+---+---+   |     +---+---+---+---+---+---+---+---+
1 |   |   |   |   |   |   |   |   |   |   1 | X |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |     +---+---+---+---+---+---+---+---+
2 |   |   |   |   |   |   |   |   |   |   2 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |     +---+---+---+---+---+---+---+---+
3 |   |   |   |   |   |   |   |   |   |   3 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |     +---+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   |   |   |   |   4 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |     +---+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |   |   5 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |     +---+---+---+---+---+---+---+---+
6 |   |   |   |   |   |   |   |   |   |   6 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |     +---+---+---+---+---+---+---+---+
7 |   |   |   |   |   |   |   |   |   |   7 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |     +---+---+---+---+---+---+---+---+
8 |   |   |   |   |   |   |   |   |   |   8 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |     +---+---+---+---+---+---+---+---+

Please select a row and a column: 8 5
```

Command Line

```
BATTLESHIPS

Stewie, it is your turn! Attack the right board.

    1   2   3   4   5   6   7   8           1   2   3   4   5   6   7   8
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
1 |   |   |   |   |   |   |   |   |   1 | X |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
2 |   |   |   |   |   |   |   |   |   2 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
3 |   |   |   |   |   |   |   |   |   3 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   |   |   |   4 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |   5 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
6 |   |   |   |   |   |   |   |   |   6 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
7 |   |   |   |   |   |   |   |   |   7 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
8 |   |   |   |   | 0 |   |   |   |   8 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+

Please select a row and a column: 2 1

BATTLESHIPS

Brian, it is your turn! Attack the left board.

    1   2   3   4   5   6   7   8           1   2   3   4   5   6   7   8
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
1 |   |   |   |   |   |   |   |   |   1 | X |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
2 |   |   |   |   |   |   |   |   |   2 | 0 |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
3 |   |   |   |   |   |   |   |   |   3 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   |   |   |   4 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |   5 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
6 |   |   |   |   |   |   |   |   |   6 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
7 |   |   |   |   |   |   |   |   |   7 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
8 |   |   |   |   | 0 |   |   |   |   8 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+

Please select a row and a column: 1 1
```

Command Line

```
BATTLESHIPS

Stewie, it is your turn! Attack the right board.

    1   2   3   4   5   6   7   8           1   2   3   4   5   6   7   8
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
1 | X |   |   |   |   |   |   |   |   1 | X |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
2 |   |   |   |   |   |   |   |   |   2 | 0 |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
3 |   |   |   |   |   |   |   |   |   3 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   |   |   |   4 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |   5 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
6 |   |   |   |   |   |   |   |   |   6 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
7 |   |   |   |   |   |   |   |   |   7 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
8 |   |   |   |   | 0 |   |   |   |   8 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+

Please select a row and a column: 5 5

BATTLESHIPS

Brian, it is your turn! Attack the left board.

    1   2   3   4   5   6   7   8           1   2   3   4   5   6   7   8
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
1 | X |   |   |   |   |   |   |   |   1 | X |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
2 |   |   |   |   |   |   |   |   |   2 | 0 |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
3 |   |   |   |   |   |   |   |   |   3 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   |   |   |   4 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |   5 |   |   |   |   | 0 |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
6 |   |   |   |   |   |   |   |   |   6 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
7 |   |   |   |   |   |   |   |   |   7 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+
8 |   |   |   |   | 0 |   |   |   |   8 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+     +---+---+---+---+---+---+---+---+

Please select a row and a column: 1 2
```

On invalid input, the prompt is repeated. E.g., if it is Stewie's turn and he provides three invalid inputs before selecting row 3 and column 3, the turn's interaction could look like:

Command Line

```
BATTLESHIPS

Stewie, it is your turn! Attack the right board.

    1   2   3   4   5   6   7   8           1   2   3   4   5   6   7   8
  +---+---+---+---+---+---+---+---+   |   +---+---+---+---+---+---+---+---+
1 | X | X | 0 |   |   |   |   |   |   | 1 | X | X |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |   +---+---+---+---+---+---+---+---+
2 |   |   |   |   |   |   |   |   |   | 2 | 0 |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |   +---+---+---+---+---+---+---+---+
3 |   |   |   |   |   |   |   |   |   | 3 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |   +---+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   |   |   |   | 4 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |   +---+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |   | 5 |   |   |   |   | 0 |   |   |   |
  +---+---+---+---+---+---+---+---+   |   +---+---+---+---+---+---+---+---+
6 |   |   |   |   |   |   |   |   |   | 6 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |   +---+---+---+---+---+---+---+---+
7 |   |   |   |   |   |   |   |   |   | 7 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |   +---+---+---+---+---+---+---+---+
8 |   |   |   |   | 0 |   |   |   |   | 8 |   |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |   +---+---+---+---+---+---+---+---+

Please select a row and a column: one
Please select a row and a column: 1 1
Please select a row and a column: 2 1
Please select a row and a column: 3, 1
Please select a row and a column: 3 3
```

## Game Result

A result screen when a player with the name Brian won looks like:

Command Line

```
THE GAME IS OVER!

    1   2   3   4   5   6   7   8           1   2   3   4   5   6   7   8
  +---+---+---+---+---+---+---+---+   |   +---+---+---+---+---+---+---+---+
1 | X | X | O |   | X |   |   |   |   | 1 | X | X | O |   | O |   |   |   |
  +---+---+---+---+---+---+---+---+   |   +---+---+---+---+---+---+---+---+
2 | O |   |   |   | X |   |   |   |   | 2 | O | O |   |   | O |   |   |   |
  +---+---+---+---+---+---+---+---+   |   +---+---+---+---+---+---+---+---+
3 | X | X | X | X | X |   |   | X |   | 3 | X | X | X | X | O |   |   |   |
  +---+---+---+---+---+---+---+---+   |   +---+---+---+---+---+---+---+---+
4 | O |   |   |   |   |   |   | X |   | 4 | O | O |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |   +---+---+---+---+---+---+---+---+
5 | O |   |   |   |   |   |   | X |   | 5 | O |   |   |   | O |   |   |   |
  +---+---+---+---+---+---+---+---+   |   +---+---+---+---+---+---+---+---+
6 | O |   |   |   |   |   |   | X |   | 6 | O |   |   |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |   +---+---+---+---+---+---+---+---+
7 | O |   | X | X | X |   |   | X |   | 7 | O |   | O | O | O |   |   |   |
  +---+---+---+---+---+---+---+---+   |   +---+---+---+---+---+---+---+---+
8 |   |   |   |   | O |   |   |   |   | 8 | O | O | O |   |   |   |   |   |
  +---+---+---+---+---+---+---+---+   |   +---+---+---+---+---+---+---+---+

Brian won the game!

Press ENTER to return to the menu:
```