Battleship  
Bonus Project

# **Tech Stack**

Python

# **User Stories**

### Total Unweighted Project Points: /85

## Main *Stories*

* **(5 points):** As a developer, I want to create a two-player Battleship game that follows the standard Battleship rules, so that two players can enjoy the turn-based game.
* **(20 points):** As a developer, I want to use a two-dimensional List to create a game board with a minimum space size of 10x10 for each player.
* **(15 points):** As a developer, I want a player to place a destroyer (space size: 2), submarine (space size: 3), battleship (space size: 4), and aircraft carrier (space size: 5) on the game board, and ensure the ships do not overlap, so that players have the ability to move their ships around as desired.
* **(15 points):** As a developer, I want each player to have their ship locations hidden from the other player, so that the game is played fairly.
* **(10 points):** As a developer, I want to present each player, on their turn, with a console print-out of the other player’s board as they know it, so that a player can see his or her hits and misses.
* **(10 points):** As a developer, I want to present each player, on their turn, with a console print-out of his or her own board with updated hits and misses.
* **(10 points):** As a player, I want the game to keep track of which ships I have completely destroyed, so that I can keep score of who is winning.

# **Checklist**

* Research the usage of **two-dimensional Lists** in Python.
* Using a **two-dimensional List**, create a battleship game board (minimum space size of 10x10) for each player.
* Allow the player to place a destroyer (space size: 2), submarine (space size: 3), battleship (space size: 4), and aircraft carrier (space size: 5) on the game board. **Check to make sure that they do not overlap.**
* Ensure that each player can place their ships on the game board and that the ship locations will be hidden from the other player. (You can assume that each player will move away from the screen when it is not their turn).
* On each player turn they should be presented with a console printout of the other player’s board as they know it. (Ships and parts of ships that haven’t been hit should be hidden, misses should be displayed. Each player should also be able to view their own board with updated hits and misses).
* The game should keep track of which ships have been completely destroyed. This is a way to keep score.

# **Setup Steps**

1. Create a new python project using VS Code.
2. Determine which classes you will need - create a UML.
3. Create your Python classes and methods to build out functionality.

# **Resources**

### **PowerPoints**

* Python OOP
* UML Class Diagrams
* Inheritance - also see [Student Hub article](https://devcodecamp.helpkit.so/definitions-examples-and-frequently-asked-questions/vUgiURdNAgva1AaaH99uBY/inheritance-/enR8mnN4bfjSWEGezDKCEi)

### **Relevant Projects**

* Any Previous Python OOP Projects (Robo vs Dino, etc)

# **End Result**

The end result of this project will be a two-player console application simulating the classic Battleship board game. Each player will be able to place their ships to start the game, then rounds will proceed with each player selecting a square to attack. The player will be informed if their attack was a hit or miss, and they should be told when they sink an opponent’s ship. Rounds proceed until all of one player’s ships are sunk. The winner should then be announced.