Battleship

# Introduction

Battleship is a two-player game based on naval warfare. The game is traditionally played on a 10x10 grid where each player controls 5 ships of varying length: Aircraft Carrier(5 Units), Battleship(4 Units), Cruiser(3 Units), Submarine(3 Units), and Destroyer(2 Units). Players begin the game by privately selecting where they want to place their ships on the grid. After each player finished, players alternate by attempting to locate where the other players ships are on the grid that he chose. Each time a player selects a location on the grid where an opposing player has a ship, it is considered a hit. A ship has been destroyed when all spaces of the ship have been hit. When all of a player’s ships have been destroyed he has lost the game.

## Summary

Length: ~530 Lines

Number of Variables: ~20

Functions: 4

Using multiple 2D arrays I was able to store the location of each ship and compare them against the opponents attack choice. My biggest issue was deciding to keep most everything in the main function. I wanted to separate everything into much smaller functions but passing arrays constantly through them was becoming much more trouble than it was worth. However with a few printing functions I was able to shorten the code quite a bit.

# Pseudocode:

*Include System Libraries*

*Namespace Standard of libraries*

*Function Prototypes*

*Main*

*Initialize variables*

*Print Menu options*

*Input menu choice*

*Switch*

*Choice 1: Continue onto game*

*Choice 2: Print Rules*

*Choice 3: Return 0 and Kill Program*

*Input player 1 & 2 names*

*Print a blank grid with function*

*Print player 1 setup*

*Input Aircraft Carrier location*

*Check if the location is valid*

*Input Battleship location*

*Check if the location is valid*

*Input Cruiser location*

*Check if the location is valid*

*Input Submarine…*

*Print Player 2 Setup*

*\*Same input for player 2*

*Do loop*

*Increment number of turns*

*Print P1’s Attack Board*

*Input P1’s Attack*

*Check if it’s a hit*

*If it is then mark it on P1’s attack array*

*Also add the number of hits P1 has successfully made*

*If it’s not then mark it as a miss*

*Then do the same for P2*

*Check if P1 or P2 have reached 17 Hits, if not continue the loop*

*Output the winner*

*Convert Row*

*Switch*

*Swap the letter to the relevant number*

*Print Ship Locations of players*

*Loop for 10 rows*

*Loop for 10 Columns*

*If there is a present ship at the location then print an ‘X’*

*Else print an empty space*

*Print blank grid*

*Loop for 10 rows*

*Loop for 10 Columns*

*Print a space and ‘|’*

*Print attack grid*

*Loop for 10 rows*

*Loop for 10 Columns*

*If it’s a hit print an ‘x’*

*Else if it’s a miss print a ‘O’*

*Otherwise print an empty space*