**User Manual For “C++ Battleship”**

Summary: This program is a version of the classic board game: Battleship. The user can choose where to position ships on a game board that is represented by a 2D array. The user can then choose "coordinates" to fire at on the game board such as 2A or 7F, and the program will check if a ship is placed at that point, and return a message saying if it was hit or a miss. The hit/miss will be reflected by a change of symbols on the gameboard. The game is played until all the players ships are destroyed, or the computers ships are destroyed.

1. The game is played on a 10 x 10 board, with x-coordinates being labeled 1-10 (10 is represented by X), and the y-coordinates being labeled A-J.

1 2 3 4 5 6 7 8 9 X

A ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

B ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

C ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

D ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

E ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

F ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

G ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

H ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

I ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

J ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

1. The program will first ask for an x-coordinate for placing your aircraft carrier, which has a length of 5 spaces. It will then ask for a y-coordinate, and then the orientation of the ship. For example:

“At which X-coordinate would you like to place your aircraft carrier? (Length of 5):” 3

“At which Y-coordinate would you like to place your aircraft carrier? (Length of 5):” B

“Enter an orientation for your ship (horizontal or vertical):” vertical

The output for this example would be:

1 2 3 4 5 6 7 8 9 X

A ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

B ~ ~ C ~ ~ ~ ~ ~ ~ ~

C ~ ~ C ~ ~ ~ ~ ~ ~ ~

D ~ ~ C ~ ~ ~ ~ ~ ~ ~

E ~ ~ C ~ ~ ~ ~ ~ ~ ~

F ~ ~ C ~ ~ ~ ~ ~ ~ ~

G ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

H ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

I ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

J ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

1. The program will continue to ask for the positions and orientations of the remaining four ships: the battleship, destroyer, submarine, and patrol boat. In the case of the ship not fitting at the inputted coordinates, or a collision with another ship, the program will ask the user to input a different set of coordinates.

A sample output:

1 2 3 4 5 6 7 8 9 X

A ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

B ~ ~ C ~ ~ ~ ~ ~ ~ ~

C ~ ~ C ~ ~ ~ ~ D ~ ~

D ~ ~ C ~ ~ ~ ~ D ~ ~

E ~ ~ C ~ P P ~ D ~ ~

F ~ ~ C ~ ~ ~ ~ ~ ~ ~

G ~ S ~ ~ ~ ~ ~ ~ ~ ~

H ~ S ~ ~ ~ ~ ~ ~ ~ ~

I ~ S ~ ~ ~ B B B B ~

J ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

1. The program will then generate a board for the computer controlled player, using random number generation to place the ships on the board.
2. The main part of the game will then begin. The user will be asked to enter a set of coordinates to fire upon. For example:

"Enter an x-coordinate where you want to attack (1 -> X):" 1

"Enter an y-coordinate where you want to attack (A -> J):" A

1. In the event of a hit, the program will return “You Hit!”, or “You Missed.”, if it was a miss. A hit is represented by a # on the board, or a \* for a miss.
2. The computer player will then fire, and the result will either be “The Enemy Hit!”, or “The Enemy Missed!”, and shot will be reflected on the players board.
3. The game will be played until either all of the users ships or destroyed, or the computer player’s ships are destroyed. The program will then return a win or loss message, and the program will terminate.