

CSC-5 Winter 2022
Project 2
Kira Hollowed
12 February 2022

Introduction and Rules

Title: Battleship

This program runs a version of the game of battleship. This game is to be played with two players, and they are not allowed to see each other's inputs. The object of the game is to sink all 5 of the opponent's ships.

The game begins by outputting the game board, which has columns 0-14 and rows 0-19. Player 1 will place all their ships on the board by inputting the coordinates of each point of each ship. It will also ask whether the ship is facing horizontally or vertically. The ship cannot be placed diagonally, overlap another ship, or be placed off the board. Player 2 will then place their five ships, and the game will commence.

Once each players' ships are placed, the players will begin to try and sink each other's ships. Player 1 will begin by inputting a coordinate on the board in hopes that it is where their opponent's ship is placed. If they "hit" a part of their opponent's ship, the game will notify the player that they hit the ship with an X. If they miss and rather hit the water, a 'O' will appear instead. This will go back and forth between the players until one player sinks all five of the opponent's ships.

Once a player sinks all five of their opponent's ships, the screen will output a "Congratulations!" message to the player that they won the game.

Playing the Game

```
CppApplication_4 (Build, Run) CppApplication_4 (Run) Version4Project2FINAL (Build, Run) Version4Project2FINAL (Run)
sh: cls: command not found
PLAYER 1's GAME BOARD
-----
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 0 0 0 0 0 0 0 0
4 0 0 0 0 0 0 0 0 0 0 0 0 0 0
5 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6 0 0 0 0 0 0 0 0 0 0 0 0 0 0
7 0 0 0 0 0 0 0 0 0 0 0 0 0 0
8 0 0 0 0 0 0 0 0 0 0 0 0 0 0
9 0 0 0 0 0 0 0 0 0 0 0 0 0 0

INSTRUCTIONS (Player 1)

You are about to place your ships. Format should be:
Direction (0:Horizontal,1:Vertical), X (top-row) coords, Y (left-side) coords
Example: 0 1 2 This would place a ship beginning at X:1 Y:2 going horizontal

Ship to place: Cruiser which has a length of 2
Place Ship:0
1
2
sh: cls: command not found
PLAYER 1's GAME BOARD
-----
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 0 S S 0 0 0 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 0 0 0 0 0 0 0 0
4 0 0 0 0 0 0 0 0 0 0 0 0 0 0
5 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6 0 0 0 0 0 0 0 0 0 0 0 0 0 0
7 0 0 0 0 0 0 0 0 0 0 0 0 0 0
8 0 0 0 0 0 0 0 0 0 0 0 0 0 0
9 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
CppApplication_4 (Build, Run) CppApplication_4 (Run) Version4Project2FINAL (Build, Run) Version4Project2FINAL (Run)
sh: cls: command not found
PLAYER 1's GAME BOARD
-----
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 S S 0 0 0 0 0 0 0
2 0 S S 0 0 0 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 0 0 0 0 S 0 0 0
4 0 0 0 0 0 0 0 0 0 0 S 0 0 0
5 0 0 0 0 0 0 0 0 0 0 S 0 0 0
6 0 0 0 0 0 S S S 0 0 S 0 0 0
7 0 0 0 0 0 0 0 0 0 0 0 0 0 0
8 0 0 0 0 0 0 0 0 0 0 0 0 0 0
9 0 0 0 0 0 0 0 0 0 0 0 0 0 0

INSTRUCTIONS (Player 1)

You are about to place your ships. Format should be:
Direction (0:Horizontal,1:Vertical), X (top-row) coords, Y (left-side) coords
Example: 0 1 2 This would place a ship beginning at X:1 Y:2 going horizontal

Ship to place: Battleship which has a length of 5
Place Ship:1
12
2
sh: cls: command not found
PLAYER 2's GAME BOARD
-----
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 0 0 0 0 0 0 0 0
4 0 0 0 0 0 0 0 0 0 0 0 0 0 0
5 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6 0 0 0 0 0 0 0 0 0 0 0 0 0 0
7 0 0 0 0 0 0 0 0 0 0 0 0 0 0
8 0 0 0 0 0 0 0 0 0 0 0 0 0 0
9 0 0 0 0 0 0 0 0 0 0 0 0 0 0

INSTRUCTIONS (Player 2)

You are about to place your ships. Format should be:
Direction (0:Horizontal,1:Vertical), X (top-row) coords, Y (left-side) coords
Example: 0 1 2 This would place a ship beginning at X:1 Y:2 going horizontal
```

```
CppApplication_4 (Build, Run)  CppApplication_4 (Run)  Version4Project2FINAL (Build, Run)  Version4Project2FINAL (Run)

PLAYER 2's GAME BOARD
-----
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
2  0  0  0  0  0  0  0  0  0  0  0  0  0  0
3  0  0  0  0  0  0  0  0  0  0  0  0  0  0
4  0  0  0  0  0  0  0  0  0  0  0  0  0  0
5  0  0  0  0  0  0  0  0  0  0  0  0  0  0
6  0  0  0  0  0  0  0  0  0  0  0  0  0  0
7  0  0  0  0  0  0  0  0  0  0  0  0  0  0
8  0  0  0  0  0  0  0  0  0  0  0  0  0  0
9  0  0  0  0  0  0  0  0  0  0  0  0  0  0

PLAYER 1, ENTER COORDINATES TO ATTACK: 1
2
sh: cls: command not found
PLAYER 1's GAME BOARD
-----
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
2  0  0  0  0  0  0  0  0  0  0  0  0  0  0
3  0  0  0  0  0  0  0  0  0  0  0  0  0  0
4  0  0  0  0  0  0  0  0  0  0  0  0  0  0
5  0  0  0  0  0  0  0  0  0  0  0  0  0  0
6  0  0  0  0  0  0  0  0  0  0  0  0  0  0
7  0  0  0  0  0  0  0  0  0  0  0  0  0  0
8  0  0  0  0  0  0  0  0  0  0  0  0  0  0
9  0  0  0  0  0  0  0  0  0  0  0  0  0  0

PLAYER 2, ENTER COORDINATES TO ATTACK: 5
6
sh: cls: command not found
PLAYER 2's GAME BOARD
-----
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
2  0  X  0  0  0  0  0  0  0  0  0  0  0  0
3  0  0  0  0  0  0  0  0  0  0  0  0  0  0
4  0  0  0  0  0  0  0  0  0  0  0  0  0  0
5  0  0  0  0  0  0  0  0  0  0  0  0  0  0
6  0  0  0  0  0  0  0  0  0  0  0  0  0  0
7  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

```
CppApplication_4 (Build, Run)  CppApplication_4 (Run)  Version4Project2FINAL (Build, Run)  Version4Project2FINAL (Run)

PLAYER 2's GAME BOARD
-----
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
1  0  0  0  0  0  X  0  0  0  0  0  0  0  0
2  0  X  X  0  0  0  0  0  0  0  0  0  0  0
3  0  0  0  0  0  0  0  0  0  0  0  0  0  0
4  0  0  0  0  0  0  0  0  0  0  0  0  0  0
5  0  0  0  0  0  0  0  0  0  0  0  0  0  0
6  0  0  0  0  0  X  X  0  0  0  0  0  0  0
7  0  0  0  0  0  0  0  0  0  0  0  0  0  0
8  0  0  0  0  0  0  0  0  0  0  0  0  0  0
9  0  0  0  0  0  0  0  0  0  0  0  0  0  0

PLAYER 1, ENTER COORDINATES TO ATTACK: 10
3
sh: cls: command not found
PLAYER 1's GAME BOARD
-----
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
2  0  0  0  0  0  0  0  0  0  0  0  0  0  0
3  0  0  0  0  0  0  0  0  0  0  0  0  0  0
4  0  0  0  0  0  0  0  0  0  0  0  0  0  0
5  0  0  0  0  0  0  0  0  0  0  0  0  0  0
6  0  0  0  0  0  X  0  0  0  0  0  0  0  0
7  0  0  0  0  0  0  0  0  0  0  0  0  0  0
8  0  0  0  0  0  0  0  0  0  0  0  0  0  0
9  0  0  0  0  0  0  0  0  0  0  0  0  0  0

PLAYER 2, ENTER COORDINATES TO ATTACK: 4
2
sh: cls: command not found
PLAYER 2's GAME BOARD
-----
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
1  0  0  0  0  0  X  0  0  0  0  0  0  0  0
2  0  X  X  0  0  0  0  0  0  0  0  0  0  0
3  0  0  0  0  0  0  0  0  0  0  X  0  0  0
4  0  0  0  0  0  0  0  0  0  0  0  0  0  0
5  0  0  0  0  0  0  0  0  0  0  0  0  0  0
6  0  0  0  0  0  X  X  0  0  0  0  0  0  0
7  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

```
CppApplication_4 (Build, Run) CppApplication_4 (Run) Version4Project2FINAL (Build, Run) Version4Project2FINAL (Run)
PLAYER 2's GAME BOARD
-----
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 X X X 0 0 0 0 0
2 0 X X 0 0 0 0 0 0 0 0 0 X 0 0
3 0 0 0 0 0 0 0 0 0 0 X 0 X 0 0
4 0 0 0 0 0 0 0 0 0 0 X 0 0 0 0
5 0 0 0 0 0 0 0 0 0 0 X 0 0 0 0
6 0 0 0 0 0 X X X 0 0 X 0 0 0 0
7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

PLAYER 1, ENTER COORDINATES TO ATTACK: 12
4
sh: cls: command not found
PLAYER 1's GAME BOARD
-----
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6 0 0 0 0 0 X 0 0 0 0 0 0 0 0 0
7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

PLAYER 2, ENTER COORDINATES TO ATTACK: 8
9
sh: cls: command not found
PLAYER 2's GAME BOARD
-----
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 X X X 0 0 0 0 0 0
2 0 X X 0 0 0 0 0 0 0 0 0 X 0 0
3 0 0 0 0 0 0 0 0 0 0 X 0 X 0 0
4 0 0 0 0 0 0 0 0 0 0 X 0 X 0 0
5 0 0 0 0 0 0 0 0 0 0 X 0 0 0 0
6 0 0 0 0 0 X X X 0 0 X 0 0 0 0
7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
CppApplication_4 (Build, Run) CppApplication_4 (Run) Version4Project2FINAL (Build, Run) Version4Project2FINAL (Run)
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6 0 0 0 0 0 X 0 0 0 0 0 0 0 0 0
7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

PLAYER 2, ENTER COORDINATES TO ATTACK: 5
7
sh: cls: command not found
PLAYER 2's GAME BOARD
-----
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 X X X 0 0 0 0 0 0 0
2 0 X X 0 0 0 0 0 0 0 0 0 X 0 0
3 0 0 0 0 0 0 0 0 0 0 X 0 X 0 0
4 0 0 0 0 0 0 0 0 0 0 X 0 X 0 0
5 0 0 0 0 0 0 0 0 0 0 X 0 X 0 0
6 0 0 0 0 0 X X X 0 0 X 0 X 0 0
7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

PLAYER 1, ENTER COORDINATES TO ATTACK: 12
7
sh: cls: command not found
PLAYER 1's GAME BOARD
-----
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6 0 0 0 0 0 X 0 0 0 0 0 0 0 0 0
7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

PLAYER 2, ENTER COORDINATES TO ATTACK: 5
```

Versions and their Modifications

Version 1

- Version 1 of Project 2 is the same as the final version of my Project 1.

Version 2

- The format for cout is altered to explain the rules of the game in more detail.
- BEFORE:

```
cout << "\n";  
cout << "INSTRUCTIONS (Player " << aplyr << endl;  
cout << "Ship Type: " << ship[ship1].name << " has a length of " <<  
ship[ship1].length << endl;  
cout << "Place ship";
```
- AFTER:

```
cout << "INSTRUCTIONS (Player " << aplyr << ")\n\n";  
cout << "You are about to place your ships. Format should be:\n";  
cout << "Facing (0:Horizontal,1:Vertical), X (top-row) coords, Y (left-side)  
coords\n";  
cout << "Example: 0 7 2   This would place a ship beginning at X:7 Y:2 going  
horizontal\n\n";  
cout << "Ship to place: " << ship[ship1].name << " which has a length of " <<  
ship[ship1].length << "\n";  
cout << "Where do you want it placed? ";
```

Version 3

- PLACESHIPS UIPlcnt()
{
 int d, a, b;
 PLACESHIPS tmp;
 //Using this as a bad return
 tmp.shipTyp.ongrid[0].x = -1;
 //Get 3 integers from user
 cin >> d >> a >> b;
 if (d!=0 && d!=1) return tmp;
 if (a<0 || a>=width) return tmp;
 if (b<0 || b>=height) return tmp;
 //Good data

```

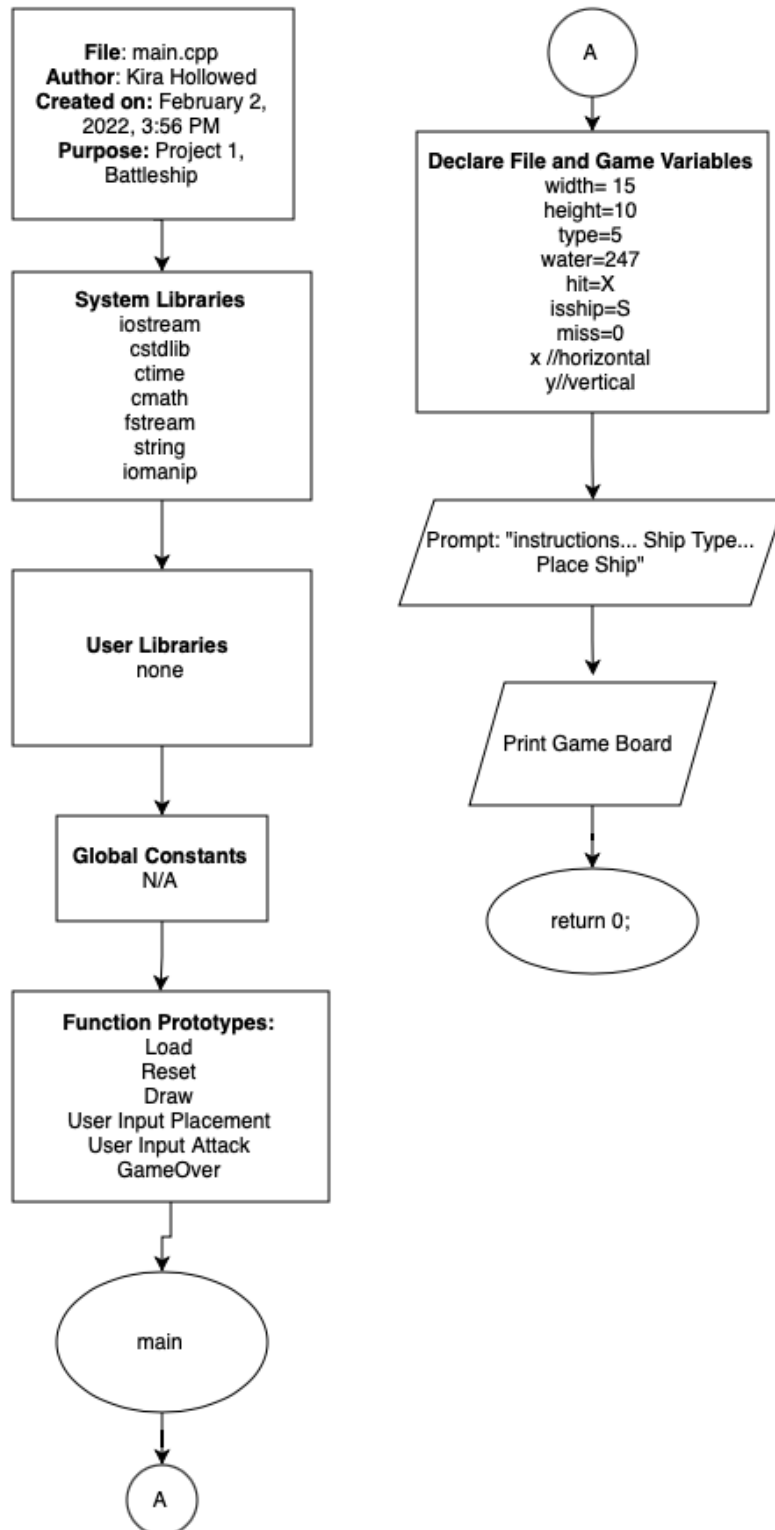
        tmp.directn = (DIRECTN)d;
        tmp.shipTyp.ongrid[0].x = a;
        tmp.shipTyp.ongrid[0].y = b;
        return tmp;
    }
    • void Load()
    {
        //Sets the default data for the ships
        //we plan to include in the game
        //MUST MATCH types -Default=5 (0-4)
        ship[0].name = "Cruiser"; ship[0].length = 2;
        ship[1].name = "Frigate"; ship[1].length = 3;
        ship[2].name = "Submarine"; ship[2].length = 3;
        ship[3].name = "Escort"; ship[3].length = 4;
        ship[4].name = "Battleship"; ship[4].length = 5;
    }
    •

```

Version 4

- Cleans up format and adds more commentary.

Flowchart



Pseudocode

```
/*
 * File:  main.cpp
 * Author: Kira Hollowed
 * Created on February 10, 2022, 10:42 PM
 * Purpose: Project 2 Version 4 FINAL, Battleship Game
 */
#include <iostream>
#include <cstdlib>
#include <ctime>
#include <cmath>
#include <fstream>
#include <string>
#include <iomanip>
using namespace std;

const int width = 15;
const int height = 10;
const int type = 5;
const float water = 247 ; //ASCII Character Code
const char hit = 'X';
const char isship = 'S';
const char miss = '0';

struct POINT {
    //A location on the grid defined
    //by X(horizontal) Y(vertical) coordinates
    int x;
    int y;
};

struct SHIP {
    //Ship name
    string name;
    //Total points on the grid
    int length;
    //Coordinates of those points
    POINT ongrid[5];
    //0-4 max length of biggest ship
    //Whether or not those points are a "hit"
```

```

        bool hitFlag[5];
    }ship[type];

    struct PLAYER {
        char grid[width][height];
    }
    player[3];
    //Just using players 1 & 2

    enum DIRECTN {horiz,vert};
    struct PLACESHIPS {
        DIRECTN directn;
        // Input 0 indicates Horizontal, Input 1 indicates Vertical
        SHIP shipTyp;
        // Cruiser, Frigate, Submarine, Escort, Battleship
    };

    bool gamerun = false;
    //Functions
    void Load(); // Board Loads
    void Reset(); // Board Resets
    void Draw(int); // Draw Board
    PLACESHIPS UIPlcnt(); // User Input Placement
    bool UIAttck(int&,int&,int); //User Input Attack
    bool GameOvr(int); //Game Over

    int main()
    {
        Load();
        Reset();

        //"PLACE SHIPS" phase of game
        //Loops through each player...
        for (int aplyr=1; aplyr<3; ++aplyr)
        {
            //Loop through each ship type to place
            for (int ship1=0; ship1<type; ++ship1)
            {
                //Display gameboard for player
                system("cls");
            }
        }
    }

```

```

Draw(aplyr);
//Give instructions
cout << "\n";
cout << "INSTRUCTIONS (Player " << aplyr << ")\n\n";
cout << "You are about to place your ships. Format should be:"<<endl;
cout << "Direction (0:Horizontal,1:Vertical), X (top-row) coords, Y (left-side)
coords"<<endl;
    cout << "Example: 0 1 2   This would place a ship beginning at X:1 Y:2 going
horizontal"<<endl<<endl;
    cout << "Ship to place: " << ship[ship1].name << " which has a length of " <<
ship[ship1].length << endl;
    cout << "Place Ship:";

//Get input from user and loop until good data is returned
PLACESHIPS aShip;
aShip.shipTyp.ongrid[0].x = -1;
while (aShip.shipTyp.ongrid[0].x == -1)
{
    aShip = UIPlcnt();
}

//Combine user data with "this ship" data
aShip.shipTyp.length = ship[ship1].length;
aShip.shipTyp.name = ship[ship1].name;

//Add the FIRST grid point to the current player's game board
player[aplyr].grid[aShip.shipTyp.ongrid[0].x][aShip.shipTyp.ongrid[0].y] = isship;

//Determine ALL grid points based on length and direction
for (int i=1; i<aShip.shipTyp.length; ++i)
{
    if (aShip.directn == horiz){
        aShip.shipTyp.ongrid[i].x = aShip.shipTyp.ongrid[i-1].x+1;
        aShip.shipTyp.ongrid[i].y = aShip.shipTyp.ongrid[i-1].y; }
    if (aShip.directn == vert){
        aShip.shipTyp.ongrid[i].y = aShip.shipTyp.ongrid[i-1].y+1;
        aShip.shipTyp.ongrid[i].x = aShip.shipTyp.ongrid[i-1].x; }

//Add the REMAINING grid points to our current players game board
player[aplyr].grid[aShip.shipTyp.ongrid[i].x][aShip.shipTyp.ongrid[i].y] = isship;

```

```

    }
    //Loop back through each ship type
    }
    //Loop back through each player
    }

    //FINISHED WITH "PLACE SHIPS" PHASE

    //Ready to play the game
    gamerun = true;
    int plyr = 1;
    do {
        //Because we are ATTACKING now, the
        //opposite players board is the display board
        int enemy;
        if (plyr == 1) enemy = 2;
        if (plyr == 2) enemy = 1;
        system("cls");
        Draw(enemy);

        //Get attack coordinates from this player
        bool goodInput = false;
        int a,b;
        while (goodInput == false) {
            goodInput = UIAttck(a,b,plyr);
        }

        //Check board; if a ship is there, set as HIT.. otherwise MISS
        if (player[enemy].grid[a][b] == isship) player[enemy].grid[a][b] = hit;
        if (player[enemy].grid[a][b] == water) player[enemy].grid[a][b] = miss;

        //Check to see if the game is over
        //If 0 is returned, nobody has won yet
        int win = GameOvr(enemy);
        if (win != 0) {
            gamerun = false;
        }
        //Alternate between each player as we loop back around
        plyr = (plyr == 1) ? 2 : 1;
    } while (gamerun);

```

```

        system("cls");
        cout << endl<< "Congratulations!!! Player " << plyr << " has won the game!"
<<endl<<endl<<endl;

```

```

        system("pause");
        return 0;
}

```

```

bool GameOvr(int oppon)
{
    bool winner = true;
    //Loop through enemy board
    for (int w=0; w<width; ++w){
        for (int h=0; h<height; ++h){
            //If any ships remain, game is NOT over
            if (player[oppon].grid[w][h] = isship)
            {
                winner = false;
                return winner;
            }
        }
    }
    //If we get here, somebody won, game over!
    return winner;
}

```

```

bool UIAttck(int& a, int& b, int theplayer)
{
    cout << endl<< "PLAYER " << theplayer << ", ENTER COORDINATES TO
ATTACK: ";
    bool goodInput = false;
    cin >> a >> b;
    if (a<0 || a>=width) return goodInput;
    if (b<0 || b>=height) return goodInput;
    goodInput = true;
    return goodInput;
}

```

```
PLACESHIPS UIPlcnt()
```

```
{
    int d, a, b;
    PLACESHIPS tmp;
    //Using this as a bad return
    tmp.shipTyp.ongrid[0].x = -1;
    //Get 3 integers from user
    cin >> d >> a >> b;
    if (d!=0 && d!=1) return tmp;
    if (a<0 || a>=width) return tmp;
    if (b<0 || b>=height) return tmp;
    //Good data
    tmp.directn = (DIRECTN)d;
    tmp.shipTyp.ongrid[0].x = a;
    tmp.shipTyp.ongrid[0].y = b;
    return tmp;
}
```

```
void Load()
```

```
{
    //Sets the default data for the ships we plan to include in the game
    //MUST MATCH types -Default=5 (0-4)
    ship[0].name = "Cruiser"; ship[0].length = 2;
    ship[1].name = "Frigate"; ship[1].length = 3;
    ship[2].name = "Submarine"; ship[2].length = 3;
    ship[3].name = "Escort"; ship[3].length = 4;
    ship[4].name = "Battleship"; ship[4].length = 5;
}
```

```
void Reset()
```

```
{
    //Loop through each player
    for (int plyr=1; plyr<3; ++plyr)
    {
        //For each grid point, set contents to 'water'
        for (int w=0; w<width; ++w){
            for (int h=0; h<height; ++h){
                player[plyr].grid[w][h] = water;
            }
        }
        //Loop back to next player
    }
}
```

```
}
```

```
void Draw(int thisPlayer)
```

```
{
```

```
    //Draws the board for a player (thisPlayer)
```

```
    cout << "PLAYER " << thisPlayer << "'s GAME BOARD"<< endl;
```

```
    cout << "-----"<<endl;
```

```
    //Loop through top row (board_width) and number columns
```

```
    cout << "  ";
```

```
    for (int w=0; w<width; ++w) {
```

```
    if (w < 10)
```

```
        //Numbers only 1 character long, add two spaces after
```

```
        cout << w << "  ";
```

```
        else if (w >= 10)
```

```
            //Numbers 2 characters long, add only 1 space after
```

```
            cout << w << " ";
```

```
        }
```

```
    cout << endl;
```

```
    //Loop through each grid point and display to console
```

```
    for (int h=0; h<height; ++h){
```

```
        for (int w=0; w<width; ++w){
```

```
            //If this is the first (left) grid point, number the grid first
```

```
            if (w==0) cout << h << " ";
```

```
            //If h was 1 character long, add an extra space to keep numbers lined up
```

```
            if (w<10 && w==0) cout << " ";
```

```
            //Display contents of this grid (if game isn't running yet, we are placing ships
```

```
            //so display the ships
```

```
            if (gamerun == false) cout << player[thisPlayer].grid[w][h] << " ";
```

```
            //Don't show ships, but show damage if it's hit
```

```
            if (gamerun == true && player[thisPlayer].grid[w][h] != isship)
```

```
                {cout << player[thisPlayer].grid[w][h] << " ";
```

```
                else if (gamerun == true && player[thisPlayer].grid[w][h] == isship)
```

```
                    {cout << water << " ";
```

```
                    //If we have reached the border.. line feed
```

```
                    if (w == width-1) cout <<endl;
```

```
                }
```

```
            }
```

```
}
```