Project 1

<Battleship Game>

CIS-5

Seth Cross

5/6/2023

Introduction

Title: Battleship Game

Battleship is a game that is played on a 10x10 grid between 2 players.
Each player will have 10 battleships of varying lengths, but only have a width of 1.
Each player will hide their board and place battleships in random positions.
Each player will take turns guessing positions where they think a battleship will be and the other player will tell them if they hit a battleship or not.
If a battleship is hit, both players will put a red marker on the spot the battleship was hit.
If a battleship is hit in every spot that it takes up, it will be considered “sunk”.
The player who sinks all enemy battleships first, wins.

Summary

Project Size: 114+ lines

Number of Variables: 7

The number of method: 2+ (not sure exactly what this means, im assuming its asking ‘how many versions’)

This project includes some of the concepts we learned from the chapters in the book. Though, since it is limited to concepts we learned from Chapters 1-5, it is incomplete. An example of a concept that could be used in a future version is arrays and functions.

It took me about 3-4 days to complete.

A problem I had faced was leaving notes of improvement as I coded up the project. That is probably something I should work on, but I find it a bit easier to just fix something up quickly and move on to the next thing. I had actually spent a few hours reworking version 1 and when you said that you wanted to see improvement, I had to go back and make new programs and put in the old mistakes that I already fixed.

The biggest problem I had was the battleship positioning, even with the do-while loop to keep randomizing the positions til none of the battleships were on top of each other. Although very rarely, some would still slip through and would be on top of eachother. It still happens in the final version, but I can’t seem to find how it is even possible as it shouldn’t even run until none of the battleships are on top of eachother.

Pseudo Code

```
Initialize

Battleship positions are randomized

Displays game starting message

Display "Input Coordinates between 0 and 20 for Missile Strike."

do
    Input a number between 0-20

    If input < 0 or > 20
        Display "Invalid Coordinates."
        Display "Input Coordinates between 0 and 20 for Missile Strike."
        Input a number between 0-20


    If input value == battleship 1 position
        *Sets battleship 5 out of range, so it can't be sunk twice*
        Display "HIT" message
        Hit counter + 1
        Display "Input Coordinates between 0 and 20 for Missile Strike."
        Input a number between 0-20
        Else If input value == battleship 2 position
                *Sets battleship 5 out of range, so it can't be sunk twice*
                Display "HIT" message
                Hit counter + 1
                Display "Input Coordinates between 0 and 20 for Missile Strike."
                Input a number between 0-20

        Else If input value == battleship 3 position
                *Sets battleship 5 out of range, so it can't be sunk twice*
                Display "HIT" message
                Hit counter + 1
                Display "Input Coordinates between 0 and 20 for Missile Strike."
                Input a number between 0-20

        Else If input value == battleship 4 position
                *Sets battleship 5 out of range, so it can't be sunk twice*
                Display "HIT" message
                Hit counter + 1
                Display "Input Coordinates between 0 and 20 for Missile Strike."
                Input a number between 0-20

        Else If input value == battleship 5 position
                *Sets battleship 5 out of range, so it can't be sunk twice*
                Display "HIT" message
                Hit counter + 1
                Display "Input Coordinates between 0 and 20 for Missile Strike."
                Input a number between 0-20

        Else
                Display "Miss" message

While (Hit Counter < 5)
Display 'Win' message.
```

Major Variables

Type	**Variable Name**	**Description**	**Location**
Unsigned Integer	bship1	Battleship 1 position	Int main()
	bship2	Battleship 2 position	Int main()
	bship3	Battleship 3 position	Int main()
	bship4	Battleship 4 position	Int main()
	bship5	Battleship 5 position	Int main()
	hitCnt	Used to keep track of “sunk” battleships, so that the game ends after sinking all ships	Int main()
Char	choice		Int main()