

Background

In class, we've been talking about pointers, references, memory management, and the symbol table. The purpose of this lab is to give you practice using these concepts in your code. To do this, we are going to create several functions that use pointers or references as parameters. We will do this in the context of a game named "Ships". Ships is a simplified version of "battleship" in which each player has only one ship. and that ship occupies a single square

Step 1. Setting Up

Create a directory named Project6. Change to that directory and create a file named ships.cc.

Initially, ships.cc should look like this:

```
#include <iostream>
#include <iomanip>
#include <stdlib.h>

using namespace std;

int main(int argc, char** argv){
    int ship1X;
    int ship1Y;
    int ship2X;
    int ship2Y;
    bool shipSunk;

    //Ask player one where his ship should go.
    getCoordinatesFromUser(ship1X,ship1Y);

    //Ask player two where his ship should go.
    getCoordinatesFromUser(ship2X,ship2Y);

    shipSunk = false;

    //While neither player has sunk a ship:
    while (!shipSunk){

        //Let player 1 guess where ship 2 is.
        shipSunk = checkGuess(&ship2X, &ship2Y);

        //If he found it, he wins.
        if (shipSunk){
            cout << "Player 1 wins!" << endl;
        }
        else{
            //Otherwise, let player 2 guess where ship 1 is.
            shipSunk = checkGuess(&ship1X,
            &ship1Y); //If he found it he wins.
            if (shipSunk){
```

```

        cout << "Player 2 wins!" << endl;
    }
}
}

```

Step 2. The GetCoordinates function

Write a function named "getCoordinatesFromUser" that returns *void* and takes two integer reference parameters named *x* and *y*:

1. Prints out a message asking the user for the *x* and *y* coordinates of his ship.
2. Read two values into *x* and *y*.
3. Checks to make sure that each number is greater than 0 and less than 15.
4. If either of them is not, then the user should be given an error message and the process should repeat (you can do this easily with a while loop).

Step 3. The checkGuess function

Now write a function named "checkGuess" that returns a boolean value and takes two integer pointers as parameters.

The checkGuess function should do the following things:

1. Ask the user to guess where his opponents ship is.
2. Read in two values (for the *x* and *y* coordinates of the guess) and store them in temporary variables.
3. Compare them with the *Values* of the variables passed into checkGuess.
4. If both values match, the function should print the message "Hit!" and return true.
5. Otherwise, it should print the message "Miss!" and return false;

Step 4. Glitter

Extend the project in some interesting way. Here are some suggestions for interesting glitter:

1. Make the ships move one space after every shot.
2. Add extra ships for each player.
3. Add ships that take up more than one square.
4. Add code for printing out an ascii art map of the position.
5. Your own original idea.