**Project 1**

**<Battleship Game>**

**CIS-5  
Name: Jorge Rubi  
Date: 11/5/17**

**Introduction**

Title: Battleship Game

Battleship is a game typically played by two opponents which possess gridded game boards hidden to one another in which the players place their “ships” privately on their respective sides.   
The opponents then proceed to take turns trying to guess the positions of their opponent’s battleships so that they may take them all down and ultimately win the game.   
Due to the nature of the game board, the grid’s labeling of letters on the y-axis and labeling of numbers on the x-axis incites players to call their guess on their opponent’s ship location with exclamations the likes of: “B4!”, “C7!”, and “F1!”.   
Thus, through a bit of luck and strategy, the player who first calls, locates, and “Hits” all their opponent’s ships will be crowned the victor.

**Summary**

Project size: approximately 200 lines  
Number of variables: 14

Through this project I attempted to create my own iteration of “*Battleship*” in C++.   
Instead of the traditional player vs player version, however, I decided to make the game more “one-sided” as the program consists of the user versus the computer.   
To hopefully better accommodate for the format in C++, I made it so that the user is against the computer, but the user is the only one doing any battleship guessing.

Although I had some previous project experience in Python, I still found it quite challenging to create the game in the manner which I desired.

Nevertheless, through my attempts at implementing concepts I’ve learned through my previous experience in Python, and through the knowledge I’ve been able to learn this semester in CIS-5, I was able to roughly create the game I desired.

With full disclosure, however, I am fully aware that there are features to the game which I couldn’t tackle in the time allotted.

Additionally, the current state of the game can be ridiculously challenging as the placement of the computer’s ships have the possibility of overlapping with one another: an error I plan to assess in Project 2, but more on that later.

**Description**

The main objective of this program is to challenge the player to guess the position of their opponent’s battleships hidden throughout the game board.

**Program**

#include <iostream>

#include <ctime>

#include <cstdlib>

#include <iomanip>

#include <fstream>

#include <string>

using namespace std;

int main() {

srand(static\_cast<unsigned int>(time(0)));

char board[10][10];

string rows,cols,guess;

short turns;

short flip;

short coordY,coordX;

bool ship1,ship2,ship3,ship4,ship5;

string answer;

rows = "ABCDEFGHIJ";

cols = " 0 1 2 3 4 5 6 7 8 9";

cout<<" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ "<<endl;

cout<<"| W E L C O M E |"<<endl;

cout<<"| T O |"<<endl;

cout<<"| BATTLE SHIP |"<<endl;

cout<<"|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|"<<endl;

cout<<"To play, enter your guess location of your opponent's ships!"<<endl;

cout<<"You can do so in a 'B4', 'F7', and 'D9' manner."<<endl;

cout<<"Good luck!"<<endl<<endl;

cout<<"To begin, please enter the number of turns you wish to play: ";

cin>>turns;

cin.ignore();

ship1 = true;

ship2 = true;

ship3 = true;

ship4 = true;

for(short temp = 0; temp < 10; temp++){

for(short tempTwo = 0; tempTwo < 10; tempTwo++){

board[temp][tempTwo] = 0;

}

}

do{

for(short y=0; y<10; y++){

if(y==0){

cout<<cols<<endl;

}

cout<<rows[y]<<" ";

for(short x=0; x<10; x++){

if((board[guess[0]][guess[1]]) == 1){

board[y][x] = {'X'};

}

else if(!(board[y][x] == 'X')){

board[y][x] = {'+'};

}

else{

board[y][x] = {'X'};

}

cout<<board[y][x]<<" ";

}

cout<<endl;

}

if(ship1==true){

flip = rand()%2+1; //[1,2]

if(flip==1){

coordX = rand()%9+0;

coordY = rand()%8+0

board[coordY][coordX] = 1;

for(short size=2;size>1;size--){

board[(coordY+=1)][coordX] = 1;

}

}

if(flip==2){

coordX = rand()%8+0;

coordY = rand()%9+0;

board[coordY][coordX] = 1;

for(short size=2;size>1;size--){

board[coordY][(coordX+=1)] = 1;

}

}

ship1=false;

}

if(ship2==true){

flip = rand()%2+1; //[1,2]

if(flip==1){

coordX = rand()%9+0;

coordY = rand()%7+0;

board[coordY][coordX] = 1;

for(short size=3;size>1;size--){

board[(coordY+=1)][coordX] = 1;

}

}

if(flip==2){

coordX = rand()%7+0;

coordY = rand()%9+0;

board[coordY][coordX] = 1;

for(short size=3;size>1;size--){

board[coordY][(coordX+=1)] = 1;

}

}

ship2=false;

}

if(ship3==true){

flip = rand()%2+1;

if(flip==1){

coordX = rand()%9+0;

coordY = rand()%6+0;

board[coordY][coordX] = 1;

for(short size=4;size>1;size--){

board[(coordY+=1)][coordX] = 1;

}

}

if(flip==2){

coordX = rand()%6+0;

coordY = rand()%9+0;

board[coordY][coordX] = 1;

for(short size=4;size>1;size--){

board[coordY][(coordX+=1)] = 1;

}

}

ship3=false;

}

if(ship4==true){

flip = rand()%2+1;

if(flip==1){

coordX = rand()%9+0;

coordY = rand()%5+0;

board[coordY][coordX] = 1;

for(short size=5;size>1;size--){

board[(coordY+=1)][coordX] = 1;

}

}

if(flip==2){

coordX = rand()%5+0;

coordY = rand()%9+0;

board[coordY][coordX] = 1;

for(short size=5;size>1;size--){

board[coordY][(coordX+=1)] = 1;

}

}

ship4=false;

}

cin>>guess;

(guess[0]>=65||guess[0]<=97) ? (guess[0] -= 65) : (guess[0] -= 97);

if((guess[1]>=48)&&(guess[1]<=57)){

guess[1] -= 48;

}

turns-=1;

}while(turns>=0);

return 0;

}