**Battleship**

**Project 1**

**CSC 5 - 45561**

**Anum Kazmi**

**07/21/17**

**Program Synopsis:**

The programs functionality is to display the outputs and insure the user correctly inputs the accurate information to successfully win the game. The program begins by introducing the user to the game and the rules the game entails. In order for the game to run the user must pick no more than 3 enemy battleships and no larger than three spaces. The board contains rows and columns of lengths of 5 spaces equally.

The first version for this game was only able to ask the user to input data and ensure that the user inputs data that follows the guidelines using Do-While loops. If the user does not follow the guidelines the program should terminate.

The second developed version of this program allows for the end results of the game. If the user guesses the enemies ships location correctly within 25 guesses, the user will be guided through the if/else statements. The if/else statements will announce whether that user is victorious and successfully completed the game or if the user was unsuccessful and lost.

**Program:**

//System Libraries

#include <iostream> //Input - Output Library

#include <iomanip> //Loops

#include <string>

#include <cstdlib>

#include <ctime> //for time function

using namespace std; //Name-space under which system libraries exist

//User Libraries

//Global Constants

//Function Prototypes

//Execution begins here

int main(int argc, char\*\* argv) {

//Declare variables

const int ROWS = 5;//maximum amount of rows for the game board

const int COLS =5;//maximum amount of columns for the game board

int nShips; //number of ships

int szShip;// length/size of ships

int nBombs=25;//number of bombs allowed per game

int iGuess;//guesses per play

int jGuess;//guesses per play

//Initialize variables

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

//Output the transformed data

cout<<endl;

cout<<"Battleship : This is a one player game and you will be able to set your difficulty."<<endl;

cout<<endl;

//input amount of ships using do-while loop

do{

cout<<"How many Battleships do you want? (You cannot have more than 3):"<<endl;

cin>>nShips;

}

while (nShips <1||nShips>3);

cout<<endl;

//input length or size of the battleship using a for loop with a nested do-while loop

for (int i = 1; i<=nShips;i++){

do{

cout<<"Size of enemy battleship #: "<<i<<"(at least 1, up to 3): ";

cin>>szShip;

}

while (szShip < 1 || szShip >3);

}

cout<<endl;

//This boolean statement is for when user misses or hits the enemies ship

bool victory = false;

if (nShips !=1)

cout<<"\nhere is the game board. There are "<<nShips<<"enemy battleships hidden here"<<endl;

else

cout<<"Here is the game board. There is " <<nShips<<"enemy battleship(s) hidden here"<<endl;

cout<<"Input row and column to decide where to place your bombs.";

cout<<endl;

cout<<"For example, if you wanted to fire the top right corner, type \"1 10\" (row 1 column 10).\n";

cout<<"You have 25 bombs. Make your First move now. Good luck.";

cout<<endl;

for (int n = 1; n <=nBombs && !victory; n++){

cout<<"\nBomb# "<<n<<", i and j: ";

cin >>iGuess;

cin>>jGuess;

victory = true;

cout<<endl;

}

cout<<endl;

//final output of the board leading to end result

cout<< "Here is the final board:";

cout<<endl;

cout<<"A \".\" is open sea.\n"<<"An \"\*\" is a hit on an enemy battleship.\n"<<"An \"S\" is a surviving enemy battleship.";

cout<<endl;

cout<<"An \"o\" is a missed bomb.\n";

cout<<endl;

//The final results of the game

if (victory){

cout<<"Congratulations, you sunk all the enemy battleships!\n\n";

cout<<endl;

}

else {

cout<<"You didn't sink every battleship. Please try again!\n\n";

cout<<endl;

}

//Exit stage right!

return 0;

}





