Project I

<Battleship>

CSC 5 – 43952

Jerson Rosales

05/14/15

**Introduction**

Title: Battleship

This is the Battleship Game

The game consists of 5 ships the player has to sink and must enter an x and a y coordinate to pinpoint where the missile is going to hit.

The user is going to be asked to enter two integers:

For example: Enter x coordinate……5, Enter y coordinate…..7,

It is either to tell you if you hit or missed the ship but careful there are only 5 ships in the 10X10 and only one point for the ship you may attack so to sum it all up there are only 5 points out of the 100 in the grid to be able to sink the ship!

This game is mostly just for fun and future updates will arrive to where there will be more points and score feature to be able to get the ships as well as keeping track of what ships have sink!

**Summary**

Project Size=147 Lines

Number of Variables =13

This project includes many concepts we have learned and it has the potential to become a very great game such as updating the score or updating the amount of ships the player would want to have and as well as adding an AI to the game to make it much more competitive.

The project took a decent amount of time to accomplish but it is very much what Skeleton of a future to come project with many more doors so that more features may come.

The game itself was simple but achieving this game with the majority of the things we learned from the book was a great challenge and this helped hone more of my programming skills to achieve a more.

Something that is not acceptable was the Global variable but it was a necessary evil to be done due to the fact that I’m not fluent or no the syntax to orient pointers in this application. But in the future I will have mastered and removed the Global Variable to add the pointer.

C++ is a very difficult language to learn but this is only the beginning and without further a do ill introduce more of my battleship game.

**Description**

A main reason I had to make this program was because I found it easier and best to utilize Arrays which was something I was having difficulty with.

**Psuedo Code**

*Main*

*Enter location for x coordinate*

*While the x coordinate is less than 0 or greater than 11*

*Enter a valid x input*

*Enter location for y coordinate*

*While y coordinate is less than 0 or greater than 11*

*Enter a valid y input*

*If the coordinates were correct*

*You Hit!*

*Else if it was wrong*

*You Missed!*

*Show Number of Ships left*

*Output Number of Ships to a file*

*Ask if they want to surrender*

*If Yes*

*Then surrender*

*If No then don’t surrender*

*Start back to Enter x coordinate unless player wants to quit*

*Game Over!*

*End Main*

**Major Variables**

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Name | Description | Location |
| Int | row | Determine row for the board | conts int row=10; |
| Int | col | Determine the columns for the board | const int col=10; |
| Int | numships | Determine how many ships are on the board | const int numships=5; |
| Int | pos1 | Determine the x coordinate of the board | int pos1; |
| Int | pos2 | Determine the y coordinate of the board | int pos2; |
| char | Board[row][col] | Makes the whole board | char board[row][col] |
| ofstream | out | Puts the number of ships to a file .doc | ofstream out; |

**C++ Constructs**

|  |  |  |
| --- | --- | --- |
| **Chapter** | **New Syntax/Keywords** | **Location** |
| 2 | cout | cout<<"Please input location:"<<endl; |
| 2 | int | int pos1,pos2; |
| 2 | char | char board[row][col]; |
| 2 | bool | bool aslt( int x,int y) |
| 2 | Arithmetic operators | else if (numShip()==0) |
| 2 | Comments | //Board |
| 2 | Named Constants | const int row=10; |
| 3 | cin | cin>>pos1; |
| 3 | Math Operators | while(pos1<=0||pos1>=11) |
| 3 | rand | int x=rand()%row;  int y=rand()%col; |
| 4 | If statement | if(aslt(pos1,pos2)) |
| 4 | If/else | if(aslt(pos1,pos2))//Assaulting the ship  cout<<"Hit!"<<endl;    else cout<<"Miss!"<<endl; |
| 4 | If/else if | if(numShip()!=0)  else if (numShip()==0) |
| 5 | Do-while | Do  while(input!='y') |
| 5 | For-loop | for (int i=0;i<row;i++) |
| 5 | Nested loops | for (int i=0;i<row;i++){  for(int j=0;j<col;j++) |
| 5 | File streaming | ofstream out; |
| 6 | Function Prototypes | void clear(); |
| 6 | Sending Data to a function | bool aslt( int x,int y){  if(board[x][y] == 1)  {  board[x][y] = 2;  return true;  }  return false; |
| 6 | Return statement | return false; |
| 7 | Array/2D | char board[row][col] |
| 7 | For-loop with Array | void visual(){  for (int i=0;i<10;i++){  for(int j=0;j<10;j++){  board[row][col]='0'; |

**Source Code**

/\*

\* File: main.cpp

\* Author: Jerson Rosales

\* Created on May 4, 2015, 2:04 PM

\* Purpose: BattleShip Game

\*/

//System Libraries

#include <iostream>

#include <ctime>

#include <cstdlib>

#include <iomanip>

#include <fstream>

using namespace std;

//User Libraries

//Global Constants

const int row=10;//Number of rows

const int col=10;//Number of columns

const int numships=5;//NUmber of Battleships

char board[row][col];//The board Variable

//Function Prototypes

void clear();//Board

int numShip();//Number of ships function

void spnShip();//Spawning the Ships

bool aslt(int,int);//The Assault on the ships

void visual();//Visual To help determine coordinates

//Execution Begins here

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

//Declare Variables

ofstream out;

int pos1,pos2;//Entering position of ships

char input;//Input for surrendering

srand(time(0));//Setting time

clear();//Setting board

spnShip();//Spawning Ships

out.open("NumOfShip.doc");//Opening file

do

{

cout<<"Please input location:"<<endl;

visual();//Visual to see the board easier

cout<<"Enter x coordinate..."<<endl;

cin>>pos1;

while(pos1<=0||pos1>=11){

cout<<"Enter a valid coordinate"<<endl;

cin>>pos1;

}

cout<<"Enter y coordinate..."<<endl;

cin>>pos2;

while(pos2<=0||pos2>=11){

cout<<"Enter a valid coordinate"<<endl;

cin>>pos2;

}

if(aslt(pos1,pos2))//Assaulting the ship

cout<<"Hit!"<<endl;

else

cout<<"Miss!"<<endl;

cout<<"Number of ships left: "<<numShip()<<endl;

//Putting number of ships to a file

out<<"The Number of ships lefts are.."<<numShip()<<endl;

//-----------------------------------------

cout<<"Do you want to surrender (y/n)? "<<endl;

if(numShip()!=0){

cin>>input;

while (input!='y'&&input!='n'){

cout<<"You must enter lower case y or n"<<endl;

cin>>input;

}

}

else if (numShip()==0){

cout<<"Enter y to quit"<<endl;

cin>>input;

while (input!='y'){

cout<<"You must enter lower case y"<<endl;

cin>>input;

}

}

}while(input!='y');

out.close();//Closing file

cout<<"Game over!"<<endl;

return 0;

}

void clear (){

for (int i=0;i<row;i++){

for(int j=0;j<col;j++){

board [i][j]=0;

}

}

}

int numShip(){

int c = 0;

for(int i=0;i<row;i++)

{

for(int j=0;j<col;j++)

{

if(board[i][j]==1)

c++;

}

}

return c;

}

void spnShip(){

int s = 0;

while(s < numships)

{

int x=rand()%row;

int y=rand()%col;

if(board[x][y]!= 1)

{

s++;

board[x][y] = 1;

}

}

}

bool aslt( int x,int y){

if(board[x][y] == 1)

{

board[x][y] = 2;

return true;

}

return false;

}

void visual(){

for (int i=0;i<10;i++){

for(int j=0;j<10;j++){

board[row][col]='0';

}

}

cout<<" 1 2 3 4 5 6 7 8 9 10"<<endl;

for (int i=0;i<=row;i++){

cout<<i<<" ";

for(int j=0;j<10;j++){

cout<<board[row][col]<<setw(2);

}

cout<<"\n";

}

}