**Final Project**

Title

**Battleship v.2**

Course

**CIS-17A**

Section

**42448**

Author

**Miguel Gaona**

1. **Introduction**

This is a revised version of the game Battleship using classes and inheritance. In this two player game, each player places five ships on a 10x10 board. Players then take turns to guess the locations of the opponent’s ships. The objective of the game is to find all of the opponent’s ships before they find yours.

1. **Summary**

Project size: about 560 lines

Number of variables: 25

Number of functions: 21

This game uses the concepts that we learned in class, such as dynamic memory allocation, classes and inheritance. The project took about a week to complete. The game could be improved by adding ships that take up multiple spaces on the board and take multiple hits to sink.

1. **Description**

I chose to make this program because I had already made a similar project on a free online learning course. The course showed me how to make a much simpler version using the Python language. I thought it might be interesting to recreate it using C++.

* 1. **Pseudocode**

**I***nitialize*

*Display welcome screen*

*Dynamically create the computer’s board(2D array)*

*Dynamically create the player’s board(2D array)*

*Place computer’s ships in random locations*

*Ask player for location to ships 1-5*

*Get locations from player*

*Display both boards*

*While either player still has a ship on their board*

*If it is player 1’s turn*

*Ask player to guess a location*

*Get location from player*

*If player’s guess equals the location of a ship*

*If ship status is “hit”*

*Tell player they already hit this ship*

*Else*

*Tell the player which ship they hit*

*Change ship status to “hit”*

*Decrement the computer’s ship count*

*Else*

*If player’s guess is not a valid choice*

*Tell player that is not a valid choice*

*Else if player already guessed that location Tell player that locations has already been guessed*

*Else*

*Tell player that they missed*

*Else if it is computer’s turn*

*Computer guesses a random location*

*Tell player what location it guessed*

*If computer’s guess matches location of one of the player’s*

*ships*

*Tell the player which ship was hit*

*Change ship status to “hit”*

*Decrement player’s ship count*

*Else*

*Tell the player that the computer missed*

*Display the resulting board after each turn*

*If the computer has no more ships left*

*The player wins*

*Else if the player has no more ships left*

*The computer wins*

*De-allocate memory*

*Exit program*

* 1. **Variables**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Name** | **Description** | **Location** |
| Const int | SIZE | Sets size for the dynamic arrays | Main(), createBoard() |
| Char\*\* | compBoard | Dynamic 2d array for the computer’s board | Main(), initializeBoard(), printBoard() |
|  | playerBoard | Dyanamic 2d array for the player’s board | Main(), initializeBoard(), printBoard() |
| Int | randomRow | Holds a random row location | Class computer,  main(), setRandomRow(),  getRandomRow() |
|  | randomCol | Holds a random column location | Class Computer,  Main(),  setRandomCol(),  getRandomCol() |
|  | guessRow | Holds the guessed row location | Class Player,  Main(),  setGuessRow(),  getGuessRow() |
|  | guessCol | Holds the guessed column location | Class Player,  Main(),  setGuessCol(),  getGuessCol() |
|  | playerNum | counter that shows whose turn it is | Main() |
|  | shipsLeft | Holds the number of ships the player still has | Class Player,  Main(),  setShipsLeft(),  getShipsLeft() |
| string | n | Holds the name of the human player | Main() |
| shipLocation | ship1 | computer’s first ship | main() |
|  | ship2 | computer’s second ship | main() |
|  | ship3 | computer’s third ship | main() |
|  | ship4 | computer’s fourth ship | main() |
|  | ship5 | computer’s fifth ship | main() |
|  | ship1p | player’s first ship | main() |
|  | ship2p | player’s second ship | main() |
|  | ship3p | player’s third ship | main() |
|  | ship4p | player’s fourth ship | main() |
|  | ship5p | player’s fifth ship | main() |
| Human | player1 | Class holding the human player information | Main() |
| Computer | computer | Class holding the computer AI information | Main() |

1. **References**
   1. www.Codecademy.com
   2. Starting out with C++ - Tony Gaddis
2. **Source Code**

*/\**

*\* File: main.cpp*

*\* Author: Miguel*

*\**

*\* Created on June 2, 2014, 7:13 PM*

*\*/*

*//Libraries*

*#include <cstdlib>*

*#include <iostream>*

*#include <ctime>*

*using namespace std;*

*//declare structures*

*struct shipLocation*

*{*

*int row;*

*int col;*

*bool hit = false;*

*};*

*//function prototypes*

*char\*\* createBoard(int);*

*void welcomeScreen();*

*void initializeBoard(char\*\*);*

*int getLocation();*

*void printBoard(char\*\*);*

*int getRandomRow();*

*int getRandomCol();*

*bool checkInput(int);*

*void destroy(char\*\*,int);*

*char\*\* createBoard();*

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

*{*

*//declare variables*

*const int SIZE=10;*

*char\*\* board;*

*char\*\* playerBoard;*

*int randomRow;*

*int randomCol;*

*int guessRow;*

*int guessCol;*

*int playerNum = 1;*

*int playerShips = 5;*

*int compShips = 5;*

*int winner = 0;*

*shipLocation ship1, ship2, ship3, ship4, ship5;*

*shipLocation ship1p, ship2p, ship3p, ship4p, ship5p;*

*srand(time(0));*

*//display welcome screen*

*welcomeScreen();*

*//set up the computer's board*

*board = createBoard(SIZE);*

*initializeBoard(board);*

*//set up the player's board*

*playerBoard = createBoard(SIZE);*

*initializeBoard(playerBoard);*

*//player places their ships*

*cout<<"First let's start by placing your ships."<<endl<<endl;*

*cout<<"Enter the row for Ship 1: ";*

*ship1p.row = getLocation();*

*cout<<"Enter the column for Ship 1: ";*

*ship1p.col = getLocation();*

*cout<<"Enter the row for Ship 2: ";*

*ship2p.row = getLocation();*

*cout<<"Enter the column for Ship 2: ";*

*ship2p.col = getLocation();*

*cout<<"Enter the row for Ship 3: ";*

*ship3p.row = getLocation();*

*cout<<"Enter the column for Ship 3: ";*

*ship3p.col = getLocation();*

*cout<<"Enter the row for Ship 4: ";*

*ship4p.row = getLocation();*

*cout<<"Enter the column for Ship 4: ";*

*ship4p.col = getLocation();*

*cout<<"Enter the row for Ship 5: ";*

*ship5p.row = getLocation();*

*cout<<"Enter the column for Ship 5: ";*

*ship5p.col = getLocation();*

*//computer's ships placed randomly*

*ship1.row = getRandomRow();*

*ship1.col = getRandomCol();*

*ship2.row = getRandomRow();*

*ship2.col = getRandomCol();*

*ship3.row = getRandomRow();*

*ship3.col = getRandomCol();*

*ship4.row = getRandomRow();*

*ship4.col = getRandomCol();*

*ship5.row = getRandomRow();*

*ship5.col = getRandomCol();*

*//display boards*

*cout<<endl<<"My board"<<endl;*

*printBoard(board);*

*cout<<"Your board"<<endl;*

*printBoard(playerBoard);*

*//FOR TESTING PURPOSES*

*//if you want to see where the computer's ships are, delete the comment marks around this block.*

*cout<<"Ship 1: "<<ship1.row<<" "<<ship1.col<<endl;*

*cout<<"Ship 2: "<<ship2.row<<" "<<ship2.col<<endl;*

*cout<<"Ship 3: "<<ship3.row<<" "<<ship3.col<<endl;*

*cout<<"Ship 4: "<<ship4.row<<" "<<ship4.col<<endl;*

*cout<<"Ship 5: "<<ship5.row<<" "<<ship5.col<<endl;*

*//continue taking turns until one player loses all their ships*

*while((compShips!=0) && (playerShips!=0))*

*{*

*//player's turn*

*if (playerNum%2==1)*

*{*

*playerNum++;*

*cout<<"It's your turn."<<endl;*

*cout << "Guess a row: ";*

*cin >> guessRow;*

*cout << "Guess a column: ";*

*cin >> guessCol;*

*if (guessRow==ship1.row && guessCol==ship1.col)*

*{*

*if (ship1.hit)*

*{*

*cout<<"You already sunk this ship!";*

*}*

*else*

*{*

*cout << "You sunk Ship 1!" << endl;*

*ship1.hit = true;*

*board[guessRow][guessCol] = '1';*

*compShips--;*

*}*

*}*

*else if (guessRow==ship2.row && guessCol==ship2.col)*

*{*

*if (ship2.hit)*

*{*

*cout<<"You already sunk this ship!";*

*}*

*else*

*{*

*cout << "You sunk Ship 2!" << endl;*

*ship2.hit = true;*

*board[guessRow][guessCol] = '2';*

*compShips--;*

*}*

*}*

*else if (guessRow==ship3.row && guessCol==ship3.col)*

*{*

*if (ship3.hit)*

*{*

*cout<<"You already sunk this ship!";*

*}*

*else*

*{*

*cout << "You sunk Ship 3!" << endl;*

*ship3.hit = true;*

*board[guessRow][guessCol] = '3';*

*compShips--;*

*}*

*}*

*else if (guessRow==ship4.row && guessCol==ship4.col)*

*{*

*if (ship4.hit)*

*{*

*cout<<"You already sunk this ship!";*

*}*

*else*

*{*

*cout << "You sunk Ship 4!" << endl;*

*ship4.hit = true;*

*board[guessRow][guessCol] = '4';*

*compShips--;*

*}*

*}*

*else if (guessRow==ship5.row && guessCol==ship5.col)*

*{*

*if (ship5.hit)*

*{*

*cout<<"You already sunk this ship!";*

*}*

*else*

*{*

*cout << "You sunk Ship 5!" << endl;*

*ship5.hit = true;*

*board[guessRow][guessCol] = '5';*

*compShips--;*

*}*

*}*

*else*

*{*

*if (guessRow<0 || guessRow>9 || guessCol<0 || guessCol>9)*

*{*

*cout << "That's not on the board!" << endl;*

*}*

*else if (board[guessRow][guessCol] == 'X')*

*{*

*cout << "You already guessed that one!" << endl;*

*}*

*else*

*{*

*cout << "Miss!" << endl;*

*board[guessRow][guessCol] = 'X';*

*}*

*}*

*}*

*//computer's turn*

*else if(playerNum%2 == 0)*

*{*

*playerNum++;*

*cout<<endl<<"Ok, now it's my turn."<<endl;*

*guessRow = getRandomRow();*

*guessCol = getRandomCol();*

*cout<<"I guess Row "<<guessRow<<" Column "<<guessCol<<endl;*

*if (guessRow==ship1p.row && guessCol==ship1p.col)*

*{*

*cout << "I sunk your Ship 1!" << endl<<endl;*

*ship1p.hit = true;*

*playerBoard[guessRow][guessCol] = '1';*

*playerShips--;*

*}*

*else if (guessRow==ship2p.row && guessCol==ship2p.col)*

*{*

*cout << "I sunk your Ship 2!" << endl<<endl;*

*ship2p.hit = true;*

*playerBoard[guessRow][guessCol] = '2';*

*playerShips--;*

*}*

*else if (guessRow==ship3p.row && guessCol==ship3p.col)*

*{*

*cout << "I sunk your Ship 3!" << endl<<endl;*

*ship3p.hit = true;*

*playerBoard[guessRow][guessCol] = '3';*

*playerShips--;*

*}*

*else if (guessRow==ship4p.row && guessCol==ship4p.col)*

*{*

*cout << "I sunk your Ship 4!" << endl<<endl;*

*ship4p.hit = true;*

*playerBoard[guessRow][guessCol] = '4';*

*playerShips--;*

*}*

*else if (guessRow==ship5p.row && guessCol==ship5p.col)*

*{*

*cout << "I sunk your Ship 5!" << endl<<endl;*

*ship5p.hit = true;*

*playerBoard[guessRow][guessCol] = '5';*

*playerShips--;*

*}*

*else*

*{*

*cout << "I missed!" << endl << endl;*

*playerBoard[guessRow][guessCol] = 'X';*

*}*

*//display the resulting boards after each round*

*cout<<endl<<"My board"<<endl;*

*printBoard(board);*

*cout<<"My ships left: "<<compShips<<endl<<endl;*

*cout<<"Your board"<<endl;*

*printBoard(playerBoard);*

*cout<<"Your ships left: "<<playerShips<<endl<<endl;*

*}*

*}*

*if (compShips == 0)*

*{*

*cout<<"Congratulations! You sunk all of my ships."<<endl<<"You win!!!!!!"<<endl<<endl;*

*}*

*else if (playerShips == 0)*

*{*

*cout<<"Looks like I won! Better luck Next time!";*

*}*

*//cleanup memory*

*destroy(board,SIZE);*

*destroy(playerBoard,SIZE);*

*return 0;*

*}*

*void welcomeScreen()*

*{*

*cout<<"Welcome to Miguel Gaona's Battleship!"<<endl<<endl;*

*cout<<"Instructions: First place your ships on your board. My ships have been placed randomly on my board. We take turns guessing until one sinks all five of the other's ships."<<endl;*

*cout<<"Your input should be between 0 and 9"<<endl<<endl;*

*}*

*char\*\* createBoard(int SIZE)*

*{*

*char \*\*board = new char\*[SIZE];*

*for (int i = 0; i < SIZE; i++)*

*{*

*board[i] = new char[SIZE];*

*}*

*return board;*

*}*

*void initializeBoard(char\*\* board)*

*{*

*for (int x=0;x<10;x++)*

*{*

*for (int y=0;y<10;y++)*

*{*

*board[x][y] = 'O';*

*}*

*}*

*}*

*int getLocation()*

*{*

*int num;*

*bool invalid;*

*do{*

*cin>>num;*

*invalid = checkInput(num);*

*}while (invalid);*

*return num;*

*}*

*bool checkInput(int input)*

*{*

*if (input<0 || input>9)*

*{*

*cout<<"Not a valid input. Try again: ";*

*return true;*

*}*

*else*

*{*

*return false;*

*}*

*}*

*void printBoard(char\*\* board)*

*{*

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

*{*

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

*{*

*cout<<" "<<board[i][j];*

*}*

*cout<<endl;*

*}*

*cout<<endl;*

*}*

*int getRandomRow()*

*{*

*int row = rand()%10;*

*return row;*

*}*

*int getRandomCol()*

*{*

*int col = rand()%10;*

*return col;*

*}*

*void destroy(char\*\* board, int SIZE)*

*{*

*for(int i=0;i<SIZE;i++)*

*{*

*delete [] board[i];*

*}*

*delete [] board;*

*}*

*/\**

*\* File: Computer.h*

*\* Author: Miguel*

*\**

*\* Created on June 8, 2014, 3:06 PM*

*\*/*

*#include "Player.h"*

*#include <ctime>*

*#ifndef COMPUTER\_H*

*#define COMPUTER\_H*

*class Computer : public Player*

*{*

*private:*

*int randomRow;*

*int randomCol;*

*public:*

*Computer(int);*

*void setRandomRow();*

*int getRandomRow();*

*void setRandomCol();*

*int getRandomCol();*

*void setGuessRow();*

*void setGuessCol();*

*};*

*#endif*

*#include "Computer.h"*

*Computer::Computer(int s) : Player(s)*

*{*

*randomRow = 0;*

*randomCol = 0;*

*}*

*void Computer::setRandomRow()*

*{*

*randomRow = rand()%10;*

*}*

*int Computer::getRandomRow()*

*{*

*return randomRow;*

*}*

*void Computer::setRandomCol()*

*{*

*randomCol = rand()%10;*

*}*

*int Computer::getRandomCol()*

*{*

*return randomCol;*

*}*

*void Computer::setGuessRow()*

*{*

*guessRow = rand()%10;*

*}*

*void Computer::setGuessCol()*

*{*

*guessCol = rand()%10;*

*}*

*/\**

*\* File: Human.h*

*\* Author: Miguel*

*\**

*\* Created on June 7, 2014, 7:49 PM*

*\*/*

*#include "Player.h"*

*#include <string>*

*#ifndef HUMAN\_H*

*#define HUMAN\_H*

*class Human : public Player*

*{*

*private:*

*string name;*

*public:*

*Human(int);*

*void setName(string);*

*string getName();*

*int getLocation();*

*bool checkInput(int);*

*void setGuessRow();*

*void setGuessCol();*

*};*

*#endif /\* HUMAN\_H \*/*

*#include "Human.h"*

*Human::Human(int s) : Player(s)*

*{*

*name = "";*

*}*

*void Human::setName(string n)*

*{*

*name = n;*

*}*

*string Human::getName()*

*{*

*return name;*

*}*

*int Human::getLocation()*

*{*

*int num;*

*bool invalid;*

*do{*

*cin>>num;*

*invalid = Human::checkInput(num);*

*}while (invalid);*

*return num;*

*}*

*bool Human::checkInput(int input)*

*{*

*if (input<0 || input>9)*

*{*

*cout<<"Not a valid input. Try again: ";*

*return true;*

*}*

*else*

*{*

*return false;*

*}*

*}*

*void Human::setGuessRow()*

*{*

*cout<<"Guess a row: ";*

*cin>>guessRow;*

*}*

*void Human::setGuessCol()*

*{*

*cout<<"Guess a column: ";*

*cin>>guessCol;*

*}*

*/\**

*\* File: Player.h*

*\* Author: Miguel*

*\**

*\* Created on June 7, 2014, 7:48 PM*

*\*/*

*#ifndef PLAYER\_H*

*#define PLAYER\_H*

*#include <iostream>*

*#include <cstdlib>*

*using namespace std;*

*class Player*

*{*

*protected:*

*int guessRow;*

*int guessCol;*

*int shipsLeft;*

*public:*

*Player();*

*Player(int);*

*void setShipsLeft(int);*

*int getShipsLeft();*

*virtual void setGuessRow()=0;*

*int getGuessRow();*

*virtual void setGuessCol()=0;*

*int getGuessCol();*

*void decrementShips();*

*};*

*#endif /\* PLAYER\_H \*/*

*#include "Player.h"*

*Player::Player()*

*{*

*shipsLeft = 0;*

*}*

*Player::Player(int s)*

*{*

*shipsLeft = s;*

*}*

*void Player::setShipsLeft(int s)*

*{*

*shipsLeft = s;*

*}*

*int Player::getShipsLeft()*

*{*

*return shipsLeft;*

*}*

*int Player::getGuessRow()*

*{*

*return guessRow;*

*}*

*int Player::getGuessCol()*

*{*

*return guessCol;*

*}*

*void Player::decrementShips()*

*{*

*shipsLeft--;*

*}*