Justin Hayward
Project Proposal and Writeup

For my CSCI 1300 project I will be making a 1-player game of battleship. The goal of the game is to locate and destroy all 4 ships the computer generates and complete the game in the least amount of turns possible.

The computer will generate an 9x9 game grid as an array and 4 ships of size 2 ,3, 4, and 5. The grid will also have an extra row and column that act as coordinate markers. Row 0 will have letters A-H and column 0 will hall numbers 1-8. Each ship will have a starting position with a random x-position and y-position. And a random heading. The heading will be either North, East, South, or West. The heading will be from which direction the ship's size will expand from the base position. I want to make sure the ship positions don't interfere with each other or the sides of the grid, but I don't want all ships to be grouped around the center. To fix this problem, I will relate the starting position to the heading. If the heading is North, the y-position will be between index 8 and index <size>. If the heading is East, x-position will be between index 8 and index <size>. Same rules apply for South and West, but inverse. I will start by "placing" the carrier first and add the smaller ship afterwards, thus allowing for the least amount of collision errors. When I place each ship, I will make sure that all spaces <size> forward from <heading> are clear, otherwise, will regenerate the starting position and heading.

After this large stage of game setup, the actual game will begin. The game takes place over the course of turns, each turn follows a specific pattern, discussed in my Board.h file. Every turn, the player will be prompted to enter a set of coordinates, followed by a space. For example, a play may type E 5. The computer will read in this input, separate it via a space, and check that coordinate position. If the player inputs a number first, or if the player doesn't put a space, the computer will prompt the player to enter a new coordinate. After an acceptable coordinate is entered, the game will check the grid for the position of a ship. I may either determine this purely from start position and header, but it may be of benefit to set up a "structure" for each ship, that will act as a static multi-spot location of a ship. If the coordinate entered == a location of a ship, that index of the grid will be replaced with an "X" and if it's a miss it will be replaced with an "O". After the coordinate is checked, the computer will print "HIT!" or "MISS" and reprint the updated game board. That ship will lose one health,and if that health reaches 0, that ship is sunk. After the board is printed, the computer will also print the remaining ships row by row. For example, if the battleship is destroyed, the game will print "ships remaining: \n Carrier \n Destroyer \n Cruiser \n". This progression of play will continue until every ship has an isSunk value of true.

I currently have 2 ideas for file IO, one of which I may choose not to implement for my final project. The first and temporary idea I have is to provide the player hints after every <x> turns pass, currently 3 is my number, but after a few rounds of playtesting I will determine if I need to change that. After every 3 turns, the game will create and write to an "intel" file the uncovered version of a random row, so that the player can see if there are any hit possibilities in that row. Every 3 turns after, the file will be overwritten. The second idea that I have is to keep a running file of High Scores. After the game is over, the player will be prompted to write their name and their name and score will be written to a file of existing scores. I may choose to read

in those scores, sort the values, and rewrite the file in the correct order, but this may take a little more thought.

I designed my two classes as board and ship. The Board class will cover all the overall game mechanics. The board, the player, the turn count, etc while the Ship class will cover all the ship-based variables. The position, size, heading, health, etc. The board will have access to view the health and info to the ship, but won't be able to directly affect those variables.

Please note that these heading files may be subject to change, as I will probably add functions to test the game in various stages of play, but the final project will be cleaned up and contain the correct requirements.