Documentation

For this project, Aaron and I decided to create battleship. We created a way to play with two players and for one player to play against a randomized computer. The rules for battleship are each player is allowed to place their five ships anywhere they want to on their grid. The ship lengths are 2, 3, 3, 4, and 5. There are 4 grids. Each player has their own grid and another grid to keep track of their guesses of their opponent’s grid. Once those locations are set, the turns begin. Each player has a chance to guess a coordinate. If the coordinate they guess lines up with opponent’s ships, it is a hit and a hit is marked on the opponent’s grid and the guessers grid that is keeping track of the opponent’s grid. Also, that player gets to guess again. If not, a miss is recorded on the correct grids. A ship is sunk once an all of the grid spaces that the ship takes up have been guessed by the opponent. A player wins once all of the opponent’s ships are sunk. This is the approach we took to writing our code. For this problem we decided to create 4 different classes in order to make our code more efficient and cleaner. These classes were Ship, Grid, Player, and location. Location is the base that everything else comes from. From location is ship. From ship and location is grid. From location ship and grid is player. Battleship is the main one with a few of its own functions, but they implement all of the classes. Location is the biggest variable because each variable determines if it is a hit or a miss on the grid which alters how the game works. Upon running our code, it will prompt you how to play.

If we had more time to work on this project, we would add more redundancies to catch the user entering incorrect information and prompt them to answer again. If we were to continue to improve this code in the future, we would like to have an actual grid and actual ships as opposed to having dashes and X’s and O’s. Very far in the future it would be ideal to use the internet to play this across different devices.

An assumption we are making is that the user knows that when placing a ship, you place it by the top spot of the ship if it is a vertical ship or by the left spot if it horizontal.