“Approximately” Weekly update sheet for CSCI 130 Project

- Work commenced on the project a table is made in javascript and the html is setup to accept inputs of positions such as A10 B5 etc. This will serve as the baseline for setting our ships positions. The main functions are not set yet this is mostly prep so that we can start reading these inputs. The input text fields are given an id so that we can access what is inside of these fields later (Joseph)
- Index.html is made. This file will of course serve to allow the players to connect log in and create a match. Various XMLHTTPRequests will be made here and functions will be checking whether or not we are logged in, registered, and if both players have connected. (Adam)
- Index.html is filled in there is a function to redirect to login and register.html as well as onload that checks if we are logged in, as well as a function that waits for the second player and sends an xhr request to see if the other player is logged in and keeps checking.

- Function insertobject is created. Though the name is a bit misleading. This function creates a table and creates the tables elements. We create a row and then create a table head that attaches itself to the row of the letters A-J as in a battlefield board. Then in each row we give it ten td’s with id numbers that correspond to the index that will be used in the array that stores our board and all the pieces such as ships, torpedos, empty spaces, etc. The array holding this is called boards. In edition work is started for the opponents board. A second array called boards2 is created to hold these similar values and the table is created in insertobject2 (uncreative name I know). (Joseph)
- All the necessary files are placed to be modified for database manipulation including leaderboard, index, game.php, setupdb.php, connection.php, users.php, register and login.html are introduced. These two files are made to look nice in addition to having the frameworks for both set up. Basic xhr connections are also set up in some of these files. Later on these database php files are created and are for the most part are set up to actually connect a lot of general backend stuff that took a lot of time to actually connect and went through various iterations (Adam)

- Game.html is given all the necessary functions that it needs before it is actually set up to connect to the database for multiplayer functions. It is given aesthetics that undergo iterations to get a good feel and design. Both tables are created the arrays are set up that will hold data to pass to the database. A fire function is made though this would undergo changes after. A timer is introduced and everything else required is added as well, such as making sure that the setup works and making sure that if an invalid position is selected an error occurs. Functions are created to read the arrays and modify the boards so we can do so once we are connected. If this project were single player the project would basically be done here (Joseph)

- The first attempts are made to connect game.html to the database. The previous functions necessary for connection are also perfected. Where as before the database php files had some problems and were more of a framework they are now functioning as needed (as much as possible before the main file game.html is introduced). The leaderboard is also done. The database is also created and the username / password are created in mysql. This marks the first time that the database is called by our code. Contact.html and help.html are also created though content is not inside yet. XHR connections are cleaned and set to work as the database files undergo the same process. A readme file is made but not filled yet. A ton of backend stuff has been completed. (Adam)

- Game.html superpowers are modified to actually do something in multiplayer, the writeup of who did what is written meticulously as Joseph recalls the dates of certain updates made during the process of this project. Help.html is filled out with instructions and contact.html is filled out. Various small changes are made to the game.html file after being connected to the database. Small aesthetic changes are made and the timer function is reimplemented because it was lost in a misunderstanding previously (Joseph)
- Game.html is changed to make sure it works with the database. At this point many functions had changes that needed to be made. A huge chunk of code is cleaned meticulously and aesthetics undergo another rehaul for simpler aesthetics that look very neat. A vast amount of information on database manipulation is required to fix various problems almost all of which at this stage were fixed by Adam's knowledge of the aforementioned topic, and leaderboard is perfected. Backend is finished and perfected and connected to the frontend to make an actual game that works, the turns are implemented and xhr requests are completed. Various helper functions are created to make reading info in the board easier and check for things such as which ships are sunk. As well as being able to end the game and start the game which was very hard to do prior to this (Adam)