EECS 448

Team #16

Design Paradigm for team #14

The design paradigms designed in this project are: Top-down functional Decomposition and Event Driven Design. Top-Down functional Decomposition is important when it comes to understanding a bigger idea through the help of smaller components. In this battleship project, their big idea was to have this game work with every component. They needed to break down each function or method into smaller categories and different files within their game. For instance, they had separate areas for how many ships to select, how the board was structured, placement of the ships, the inputs, and outputs of the game. The project was divided into many files which include executive.js, render.js, globals.js, index.html, and style.css. Each file had its own purpose and therefore making it an architectural design by piecing together the software features into selective members.

It is also clear that Even Driven Design was utilized. This is a functionality presented in the program since it relays actions throughout the loosely coupled software elements. For instance, when the opponents board is empty, the player will win the game. The game will display which player has won through a png file. Essentially, the providers act as the software that is detecting the movement of the program and the consumers are the players who are choosing what occurs. To run through the program, the message channel will provide the code that is necessary to update each step. Adding new features to the game makes everything easier because of this functionality. The message channel ensures that all the updates are feasible and less likely to fail. All in all, it is straightforward to make additions when it comes to providers, message channels, and consumers.