GO Paradigm

When developing the solution in GO I opted for using a functional coding paradigm. The solution was developed by using a series of functions to separate and simplify the code. Functional programming focuses on declarations and expressions rather than the execution of statements, functions can pass arguments, and return other functions. As loops and conditional statements are less supported it was important to encapsulate all the data within functions. Such as having the display board wrapped in a function, making it easy to call on at any time, and again not worried about the logic of loops and having them nested in each other. It also greatly reduced the amount of code, as the generation of ships could be wrapped in a function and implemented that way, making it easier to again isolate errors in the code. Many comparisons were also wrapped within functions for easier access, such as checking if any of the tiles matched each other and checking for player hits. By reducing the amount of repeated code it makes it a lot more readable and easier to understand. Functions can also be isolated and easily tested for errors making the development process a lot easier too.