EECS 448  
Team #2

Design paradigm for team #12

The design paradigms designed in this project are: Top-down functional Decomposition, Object-Oriented Design, and Event Driven Design. Top-down functional Decomposition is useful when breaking down a big picture. In this project it is evident that their big picture was a battleship game with every working component. This was then broken down into smaller sections and different files in their game for selecting how many ships, the board structure, placement, the inputs, and outputs of the game. The project was broken into multiple files with each having its own purpose thus making an architectural design. Breaking the software elements into components. It is also clear that Object-Oriented Design was attempted in this program for the different components of the game. However, there were many functions that could have been condensed and done in a neater way. All in all, the attempt for this design paradigm is there.  There is a class for pieces, the board, and the structure of the code shows a skeleton for the required concepts like class diagrams, determination of formats of attributes, and methods being assigned to relevant classes. Which is why when we took control of the program, we were able to develop this further. Another paradigm is Event Driven Design. It is a functionality in the program because it transmits events among loosely coupled software components and services. When a ship is sunk its location has been marked as sunk. When a player wins the game displays the winner and the game can be restarted. In a sense the providers are the software that is detecting the flow of the program and the consumers are the players that are selecting and choosing what happens. The message channel is the code that runs through the program and updates each step. By having this functionality, it makes things easier when adding new features to the game. The message channel that updates each possible outcome of the game is less likely to fail. It is easy to add more providers, message channels, and consumers.