

Project 2 - Software Architecture

Prompt: Identify the software architecture you think the Project 1 team used and explain why you think that (250-350 words – 5%)

The software architecture that best fits the project work is Event Driven Architecture (EDA). In EDA, an 'event' is a state change; the 'provider' is responsible for detecting and transferring events; the 'message' transmits information from the 'provider' to the 'consumer'. The 'consumer' reacts to the 'event'.

This software has five files to run the battleship game: Battleship.css, Battleship.html, GraphicsUI.js, MainGame.js, and ShipPlacement.js. These files are loosely coupled together, as in an EDA system. The user interface (UI) display is controlled by GraphicsUI.js, which is displayed by Battleship.html and uses elements for Battleship.css. The html file contains different UI states for different phases of the game: start menu, player boards, and game end. Clicking buttons activates functions declared in the GraphicsUI.js file. Each button click constitutes an 'event'. For example, clicking a "start game" creates an event signal; then, the UI is changed to the ship placement board and allows Player 1 to place their ships. The gameplay is progressed by clicking buttons, so the GraphicsUI.js acts as the 'provider'.

All the front-end player interaction is controlled by GraphicsUI.js. The other JavaScript files contain the backend logic for the game: ShipPlacement.js controls user ship placement logic and MainGame.js controls player turns and end game. These files act as the 'consumer'. The software does not have any classes to contain player or game data. The data is stored in their relevant file. So, information must be shared between files through 'messages'. For this software, messages are transmitted through function parameters. For example, selecting a tile to hit calls a function that has a message to update the player boards.

EDA is the best software architecture fit for this project. Client-Server architecture is not a good fit because there are no request-response patterns within the software. This project is not N-tier architecture, because there is no data tier. This project is not Pipes-and-Filters architecture because there is bidirectional communication. One could argue that this software has peer-2-peer (P2P) architecture; However, P2P is used on larger scales with multiple machines. This project is run on only one computer with only a few files.