Sprint 1 - Code Review

Daniel Takyi, Jiwoun Kim, Mason Lane

13/11/2019

Throughout our first sprint, we endeavored to keep our code cleanly organized and easy to read. However, we also relied on various tutorials to achieve our goals. As a result, some functions were difficult to modify or clean up, as we feared causing serious damage to the reliability of our app. It is also important that we consider our variable naming conventions, as our naming quality (at the moment) is lacking. For example, ‘ul’ and ‘li’, which were used for the toolbar and its elements are not good variable names. Despite this, we believe our code is tangible and easy to work in; containing adequate comments, and space between lines for organization’s sake.

In future iterations we hope to establish permanent, quality-named variables and reorganize our functions to better accommodate the conventions taught in class. For instance, there are several methods that could be split into separate functions. Despite being relatively easy to work with, we need to work harder at keeping our code unbrittle. I believe brittle components of our code can be attributed to our lack of understanding in regards to how we communicate with the Google Places API. Since we’re new to this kind of development, our interactions with the API are disorganized and could use some improvement. We’re sure there are many effective techniques for gathering results that we simply haven't discovered. We hope to utilise these techniques as we further organize our code.

We must also split our code into several JS documents. This is simply another step towards good, modular code. Our methods must be split into smaller components, as we can assign those methods to other documents as needed. Our code *does not* currently follow the envisioned MVC architecture. Primarily, our JS file should be split into 3 or more JS files. One should cover for user input (generate button event, etc), another for data acquisition based on controller input, and a 3rd for displaying the map/data. In breaking our code apart into manageable chunks, we are hoping to also discover redundancies and eliminate them.

extra notes:

Our code spacious and organized. Throughout this first sprint, we all tried to keep our code as neat as possible. This meant adding spaces to make the code more visible. We also commented on various lines to help code viewers understand the code. We reviewed our code at the end of our sprint, ensuring that at the very least its lines were tangible and easy to follow. Despite being on the same file, we did split functions according to purpose. Our top most functions would reveal the popup and ask to initiate our map. Our middle code fetched the data, then displayed it.