Sprint 1 - Code Review

Daniel Takyi, Jiwoun Kim, Mason Lane

13/11/2019

Throughout our second sprint, we struggled to maintain or organized momentum as we approached the deadline. Our code needs to be thoroughly cleaned up. It was one of the only aspects of our kanban we could not complete by milestone 2. Our variable naming conventions have changed, in that they are more specific to what prupose they serve (see slider options). However, the very same example introduces redundancy, as several variables may be found with the same purpose/value. Despite our disorganization, our app appears reliable and ready to show off to an end user. Future updates should be dedicated towards improving the organization of our code and UI. No new features recommended until we have an effective foundation to begin anew.

Our CSS sheets have more redundancies than last review. Both the .js and .css file should be heavily analysed for redundancy. Many css classes modify, or control the same aspects when they do not need to. This is in part of our use of tutorials. Since we have relied on tutorials in some aspects, its hard to understand what will break if we change it. As such, many tutorial CSS classes we’ve used have gone unchanged, made for the specific use-case the tutorial presented. This is not acceptable.

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 had hoped to utilise these techniques for milestone 4.

In general, I believe the .js and .HTML files to be the easiest to organize. Particularly, the .js file is very disorganized, but I suspect moving some functions around could quickly improve the flow and ultimate organization of the code. I believe our code would be at an acceptable standard if we simply reorganized some of our variable/function locations (not changing algorithms, etc)

We must still split our code into several JS \*and\* CSS 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. We could not achieve this in milestone 4 - comments are also lacking from before; this can all be attributed to time-crunch.