Reactions

Liz Gooch

I drew my principle learning from overcoming the three major hurdles of Big Project: (1) what is a good candidate for an app that employs intermediate level programming and data structures?, (2) what are the components of a working program?, and (3) what details, including guarantees, need to be worked out to have a functioning complex app? At the beginning of the quarter, I could not imagine how we would work through these obstacles. And now, on the other side, I understand the process much better.

Secondly, through my work on the GUI, I found a use for GUI development in Big Project beyond technical skills. The GUI is the presentable face of the program and from the perspective, difficulties on the GUI-side can highlight decisions that need to made deeper in the program construction.

Margaret Zick

This quarter's course work has been the first time I have had to work on a coding project with peers (spanning more than a week or two or very basic in nature) since undergrad 10 years ago. The process reminded me of how different each of us mentally process and work through the projects whether that be a preference for compartmentalizing the code and working it individually or co-coding on Teams. Trying to manage our individual coding strengths and conflicting schedules and requirements made for a good exercise in creativity to maximize our individual contributions to the project.

William Frazier

Exploring programming with teams across several files with various objects and classes on a relatively small project was a great learning experience. I learned GitBash functionality and that conditions for my best productivity in a group project are different from others'.   This was humbling and helped me learn to also focus on providing materials that elevate the productivity of my team.

The MyDictionary data structure was surprisingly inefficient for extremely large data sets. In our case, we began with 400,000 clue words that resulted in 400,000 keys with a value of a list of 50 float values. Although hashing speed was similar to that of the Python dictionary, the initialization time wasn't comparable. This was reduced to 25,000 words, which was still too tasking to make a usable program with the MyDictionary data structure.

Creating classes with attributes, behaviors, and an idea of ensuring guarantees had become second nature by the time we began writing the program which was rewarding. While our team divided the workload into GUI, data structures, and functionality, combining our portions was nearly seamless.

Although not implemented in the final product, our extra feature included an autocomplete function in the GUI for valid CodeName words. The course textbook, "A Common-Sense Guide to Data Structures and Algorithms" was extremely useful for producing valid words based on any given prefix. Unfortunately. PySimpleGUI includes a word prediction method. Trying to replace this method with our own resulted in event detection complications.