Emmitt Brandt

Week 1 (Jan. 18 - 23)

The first week back was largely spent reacquainting with the team and re-familiarizing with the second semester work statement. The team met to discuss our upcoming meeting schedule, determining that the 3:00 PM Sunday meetings would still be optimal for our group as a whole. I read over the second semester work statement to ensure I remembered what was ahead and took another look at earlier documents to refresh my memory on the project so far.

Week 2 (Jan. 24 - 30)

With initial formalities and the like out of the way, I began re-examining my code from the previous semester. Making sure I recalled how it worked and had not forgotten any potentially important details. Thankfully I had pseudo-code in order to ensure neither my future self nor my team when they might need to read over it would have difficulty figuring it out. A few details needed mental refreshing, but nothing truly problematic occurred. The team meeting this week mainly aimed to ensure that each individual knew what they would be doing for the upcoming semester. I will be partially in charge of programming along with Chief. My role will largely involve keeping the code updated as I am the only full CS major on the team.

Week 3 (Jan. 31 - Feb. 06)

This week was dedicated to researching some of the functionality I wanted to add into the program that I wasn't entirely sure how to work with. I spent my time this week tracking down functions that would help with what I wanted to do. Specifically, I was looking to import new fonts into tkinter and add the "click-through" functionality I wanted. Specifically, the latter means that I wanted the onscreen keyboard to not interfere with ability to use the below software by allowing clicks on it to interact with the below window. I found exchanges and thoughts on both of them, and saved them for later. I also downloaded the modules being used, pyglet and pywin32. The first is another GUI editing tool, and the other is a windows-specific library for allowing adjustments to windows that might otherwise be impossible. During the team meeting, deadlines were set and final divvying of roles was completed. I now have more precise timeframes I need to work in for each program and each thing that might need doing.

Week 4 (Feb. 07 - 13)

This week was spent attempting to actually implement my discoveries from the previous week. Pyglet turned out to be rather simple to use. It is specifically designed with the ability to install .ttf files for various typefaces to use within Python. A version of Garamond was found on wfonts.com without issue, and was integrated without much trouble. It is now usable for the project and any further typefaces should be simple to add in the same manner if they are not supported by default. The contrast changing setting from the prototype's program can easily be edited to also change the typeface and/or text size, and will be edited for that in the coming week.

The click-through problem is proving more difficult than I expected it to be. I have found other programs that have similar compatibility to what I am looking for, but I have yet to fully understand pywin32. I can in theory make it work, but I do not believe it would be a good idea to do so before finishing breaking it down. This should not take much longer, but has proven more difficult than expected. Note has been taken of all sites looked at for crediting purposes if anything from them winds up being used in the longer term.

Week 5 (Feb. 14 - 20)

In the previous week, I had expected the click-through issue was nearing a solution. I was mistaken. Understanding of Win32 has been slower to obtain than I had expected, but that was not the whole problem. Finding what the problem was involved delving into types of extended windows and extended styles quite deeply. I tested several types of window for this purpose, and drew a blank on most of them. A transparent layered window wound up being the only feasible solution. Though even once that was discovered, the problems kept coming. The window would often re-become opaque and/or lose the capacity to send clicks to the program behind it.

Another problem with tinkering with a topmost, non-interactive window is that if your click is not sent through properly, it can entirely block the capacity to use the computer. Even swapping to a different desktop window did not help, and the task manager wound up trapped behind as well. In the end, it took several attempts ending with a twelve-hour day of troubleshooting, but the problem was finally solved.

Week 6(Feb. 21-27)

This week was dedicated to adding the first missing accessibility feature for the software - the ability to resize the keys contained within the window. In place of a static, hard-coded place for every widget, the code needed to be tweaked so that adjusting a single variable would shift everything to its new position and adjust its size. But before

that could be done, there was a first problem to solve. The sizes in the file were only two to eight. This was clearly not measured in pixels, and worse yet the program would not allow use of floating point variables in those slots. A check of the function revealed that they were measured in characters, and the errors stated it only took integers.

The solution was to detach the text on the labels from the size of the labels. I placed inside each label a totally blank image. This meant the label's contents had no size, so it would default to the minimum one pixel per unit. Once that was done, the labels could be set to place the text relative to the center of the image - not the label the image is on.

Even once that was done, dozens of lines of code needed to be edited, as every size I had was now invalid, as well as not programmed to be resized. Though that was not too terribly difficult, as it was mainly a matter of seeing what sizes best fit.