My experience with project three is really something. By itself, this is not a difficult to complete project. When this project is coupled with other class’s projects, however, it quickly becomes a nightmare experience.

In addition to this class’s project, I also had to complete two Database Systems projects, and one Computer Animation project. I was legitimately unsure I could finish this project as well as the other class’s projects on time. But I’ve at the very least managed to finish this class’s project, for better or for worse. It only took literally four all-nighters in a row to do.

Coming into the project, I believed I needed two functions and two classes at the very least. A write function, a read function, a class for the disk, and a class for the user interface. Soon into my coding, it became abundantly clear I needed much more than that. What was once only two functions swelled into around twenty. What was once two classes, grew to around eight.

In my head, and in practice, I structured my code such that all class functions were in their respective class headers. I only had one C++ file; inside it nothing but if statements to move the user around. I had one header file that contained all the include statements, enumerations, forward class declarations, define statements, and structures for all other files to have access to. The rest of the header files were divided by important classes. The Disk class and all related functions had their own header file. The file allocation types each had their own file as well. One for Contiguous Allocation, one for Indexed Allocation, and one for Chained Allocation.

As for the logic of my actual code, it was simple to understand in my mind but became difficult to implement in practice. The user simply needs to pick an option, and then the corresponding function would trigger. That was my initial thought. The result however has the user picking a choice, and then depending on that choice, it may be as simply as calling a function and just displaying stuff, or it may be as complex as having to take a pointer to the table array and passing it through consecutive function arguments to get it to the function we need it in.

The hardest part for me I found to be was passing tables and objects to the other consecutive functions. It involved using pointers which proved momentum stopping. Before, I was completing two or three functions a day. Then when I started trying to get pointers to work in my program, just trying to figure them out flushed twenty-four hours down the toilet.

Overall, what I’ve come to learn from this is how file allocation systems work. I’ve also learned of good uses for C++ structures and enumerations. I managed to also learn that trying to work on three projects at once, let alone four, is nightmarish.

At the end of the day, I don’t think my program is perfect. Had I not been under severe time constraints, I think I could have and would have found better routes for coding certain aspects of the program. But what’s done is done and I need to finish the rest of the projects whose deadlines are too close for my comfort.