Final Takeaways from the Honors Project

1. The Comfort Zone Can Only Persist for So Long

Part of the difficulty of this project was that it was the first major programming project I've ever completed with very little guidance on what to do. While I have had experience in programming from high school, most of the projects I did were small and took well under a month to complete. Most of the time, the assignment was laid out and planned for you; as long as you had a basic understanding of object-oriented programming, you should easily be able to complete these assignments without breaking a sweat.

For the first few semesters here, the pace of the introductory classes was perhaps easier than anticipated. CSE 1729 and 2050 as an experienced programmer felt like a reiteration of what I've learned in the past, with only a few new algorithms to add to my repertoire. Assignments lasted at most one and a half weeks and were no more than a few functions of code, never exceeding several hundred lines unless you were brute-forcing the assignment. Overall, nothing prepared me for the massive time commitment, organizational skills, or complexity of this project. It was the first and quickest thing I had to accept: that I had to get used to this different style of work.

Once that was internalized and out of the way, a second major roadblock arose. With no actual side projects that gained traction, coming up with a project that I would be interested in took me a great deal of time. Brainstorming something practical enough for a multithreaded or server-based project is difficult when your interests revolve around high-complexity games and immersive fictional universes.

Now that the project is over, I should confess that I don't even *like* sudokus. If you asked me, my idea of a quality puzzle game is something with largely obscure puzzle mechanics that you can find browsing swaths of indie games - something fresh, different. But several key factors led to choosing this as my honors project. First, it would make my sudoku-loving mother happy, and while I never consulted her for help for the project, it served as a baseline source of inspiration. Second, I disliked it less than chess. And third, it sounded relatively simple algorithmically.

The last point turned out to be very, very wrong.

2. Proper Planning is Hard for Large-scale Projects

One of my biggest issues as a student and as a person is my inability to plan. With a lack of experience and unreliable concentration, I had no concept of *time*. How much could I possibly complete in three months? I had no idea. I've never tried anything like this before, and if a project was spread out over several months, it was relatively easy and could be completed in a week. Reassurance from an experienced professor is one thing, but genuinely *convincing myself* of what I could accomplish in such a seemingly small timeframe is challenging. It doesn't help

that I don't even fully know what I'm getting myself into; I barely understood the concept of threads, and as much as I would've liked to have read ahead and understood it, it simply isn't something I could motivate myself to do.

Writing this report in hindsight, it's become clear that I bit off more than I could chew. While I've accomplished the baseline multithreaded algorithm, I didn't meet the things that I wanted to meet initially. Was it possible to accomplish everything that I set out to do? Possibly, if I had dedicated adequate time to the project. But these past few months I've learned a lot about what I presently can and cannot do with all factors included. I'll be able to take this information for the next time I have to undertake a major project.

3. Organization is the Most Important Part of Just About Anything

Once I stepped out of theory and stepped into programming my algorithm, I began to realize that simply *making* a sudoku isn't a black and white process. I couldn't make an NxN grid and call it a day - each cell had to have its own variables, and the sudoku itself needed to be able to categorize and analyze these guesses to draw inferences for the rest of the board. It was during this process that I began to think seriously about one of my greatest banes - organization. After several days of trying to code it out all in one compact file, I found myself spending a great deal of time reviewing my code to realign myself with it. Because of how large of a slowdown this is, I began to look into ways I could keep myself on track.

To solve a sudoku, I would need a lot of functions and several different structs, so I looked into header files. I had to break up my code into files that needed to be compiled a certain way to have all the sufficient libraries to pass the compiling process, so I looked into Makefiles. Working through the university-designated VM removed my ability to open a visual Emacs editor (as far as I figured out, at least), so I needed a way to remember every command that I found useful, so I made myself a bookmarked document. Eventually, as I went further into the code and made more and more functions, I began to lose myself in the files, so I made another document that listed out all of the functions and their purposes. If I had to list any skill that I've learned about or picked up while doing this project, it would be this - the compilation of the most essential thing anyone needs to survive, information. Having all of the information I needed right at my fingertips made the programming process much smoother, and allowed me to easily build off of my own progress. It's like notetaking, but for my thoughts and internal reasoning.

4. College is Serious and Needs Time Investment

Part of the journey in this project doesn't have to do with the project itself, but the circumstances surrounding it. The first round of midterms was the lowest point of my entire college career thus far. One could describe it as the stars aligning to completely ruin my week. My first exam was a statistics exam where I forgot my calculator. My second was Macroeconomics, arguably the class I struggled in the most. Coming out of those two exams packed into a busy Thursday didn't do well on my mental health, but I thought to myself at the time that I at least had two easy exams the next day. So when I wrapped up my first C exam by

completely bombing it, I learned the hard way that just because 1729 and 2050 were passed effortlessly, doesn't mean I can ignore my responsibilities as a student and expect myself to do well. Perhaps this was what shocked me out of my post-lockdown sluggishness, or maybe this was the first real consequence I had to face for my laziness. I am eternally grateful for the second chance at a better grade, as well as the help in regaining my confidence as an aspiring programmer. If it wasn't for the promise of a second chance, this project might have never been completed.

5. Time Management Still Needs Work

Throughout this project, I had a rather unhealthy work ethic when it came to allocating time for coding. Because there was so much time involved in this project, it gave me plenty of excuses to put it off. If there weren't weekly meetings and deadlines that we set together, I would have pushed the entire thing to the last month. Oftentimes throughout the week, from Monday to Thursday, every time I thought to work on the project, I made an excuse that there were better things to do. And there were - assignments that were due sooner, studying that needed to be done, etcetera. However, simply not allocating any time at all, not even a measly thirty minutes, was a bad habit that unfortunately is still a work in progress. 90% of the work completed for this project was done on Fridays and the weekend, and Sundays were often spent awake until 1 or 2 AM.

6. If (Backtracking == Cool) printf("Yes.\n");

In terms of things that I've learned academically beyond the scope of the class, the biggest and most useful thing I've learned was backtracking. I could say that I've grown intimately familiar with memory allocation and threads after this assignment, but backtracking became the backbone of my project and took me the most amount of time to implement. Learning and exploring different techniques of keeping track of rapidly changing data was one of the highlights of this project and is something I can carry on to other classes and future jobs.

7. Conclusion

This project has not only been extremely fun, but it opened my eyes to a lot of the skills that I still have yet to develop. It's hard to say it furthered my interest in programming, as I can confidently say that programming is something that I'm already deeply interested in. So all in all, the biggest gains are the skills that extend beyond programming - the life skills. From organizing to planning to understanding my responsibilities, I think I've learned more about myself and what I can improve on than anything else. If I had to do it all over again in a later class, I would. And when I do, I will be prepared.