Jonas Bull

Prof. Sawin

CISC 480

5/17/21

Signature Work Portfolio

The work contained in this portfolio is a miniaturization of my UST experience. It's a representation of the different challenges I faced throughout my Computer Science major, the broad experience I gained through core requirements, and the general college ethos that I developed through four years of work. The selected projects cover what I would describe as the two extremes of the Computer Science major, and the other inclusions are a simple and broad reflection of how moral and critical thinking are an essential component of the education St. Thomas seeks to serve. Throughout my time at St. Thomas, I felt both challenged to expand my understanding of the world around me and comfortable to grow and pursue my passions in both a professional and perosnal capacity. This portfolio is an expression of that belief.

I did not immediately commit to a Computer Science major. I was initially in the Physics department, and spent the first two years primarily pursuing general credits as I realized that it wasn't quite as good a fit as I'd hoped. I had also dabbled in creative writing, but I didn't find myself invested enough in the courses I'd taken here. When the time came to choose a major, I picked CS less out of some specific career goal and more because it aligned with my personal passions. I've always loved games, computers, puzzles and logic problems. I'd tried and failed to learn programming several times, though the basics were still vaguely familiar to me, and so it

seemed like the most natural fit. Evidently, it was a good choice, as it helped me to better understand why I cared about the things I did.

The earlier coursework for the CS major is not represented in this portfolio because Computer Science is a progressive field – much of the initial content is foundational and is expanded and reinforced by subsequent courses. As a result, I've selected two projects from the latter half of my major to represent what I see as the two foundational aspects of practical programming – the structured and tested "component of a whole" programming style represented by my Artificial Intelligence implementations, and the unwieldy project-management of a self-scoped program, represented by my File Explorer implementation. Both of these projects serve to expound my belief that the foundational logic of programming is a good equivalence for the variety of situations one can encounter throughout their life, particularly in a professional capacity. Understanding your role, scoping your contributions, careful planning and good execution are fundamental aspects of nearly every career, and the immediate feedback of a programming project allows for a miniaturized version of the broader processes that dictate work. This is how I believe programming has furthered my contribution to the UST mission statement and the common good as a whole, through the relation of simple programming to life's greater structures.

I will start by elaborating on the importance of the AI projects. These were a series of six average-sized projects (for which I have provided a sample of the two most significant) wherein the professor provided an existing "skeletal" implementation of some AI problem, and a suite of tests to ensure that our component functioned as intended. We were given a clear and straightforward task with known limits and asked to implement a clever solution to a problem with many avenues. This encouraged both good fundamentals and careful observation, as

oftentimes the most functional solution to the problem could be derived from thoroughly reading the instructions and understanding precisely what they've asked. Throughout my experience as a programmer, particularly in the context of personal projects, one of the most valuable skills and quickest routes to a succesful implementation is to break the problem down into as clear terms as possible, create boundaries, and solve that problem with whatever you are capable of using. Logic problems or real-world issues are often vague and overwhelming, and it can be difficult to parse what needs to be done in order to translate that into a program. The process of defining a problem in fundamental terms is critical to good programming, and the documentation and process of these AI projects helped immensely in honing that skill.

They also helped develop my ability to act as a distinct portion of a pre-existing project. We were given free access to the code for the entirety of each project, and were allowed to examine as much or as little as we needed to in order to produce a succesful product. In the industry, most if not all work is done in pre-existing systems and so the ability to examine and work directly with other people's code is essential. It also reflects the process of working in pretty much any occupation – you must first understand the scope of your specific role and how it interacts with the greater whole in order to do the best possible work. The inflexibility of the structure is essential, as adaptibility is mostly born from constraint. Through these AI projects and similar coursework, I developed and honed an essential skill for real-world contribution and functionality – the ability to comprehend and function within a system elegantly and productively.

The other CS project I have provided is the opposite – we were given a tremendously narrow chunk of starter code and asked to fully implement a graphic filesystem viewer essentially from scratch. This is a classic case of broadly executing on in-class instruction, but it

also represents the other end of the CS spectrum and how most of society functions – the process of creating the structure. After all, every system we use, from governance to schooling, had to be coined by someone. The course for which this project was made provided many techniques and suggestions for how to design and implement the software, but the significance was in the ability to take an incredibly broad possibility-space and collapse it into practical parts. This is Project Management 101, a skill that is applicable anywhere and everywhere. It's all about planning before you act, as particularly in software it's a recipe for disaster if you start coding without knowing where exactly you're going first. As this was a partnered project, it emphasized the importance of communication and alternate viewpoints – the lifeblood of a well-concieved project. Much of the project's work was comprised of studying the rubric and pre-existing solutions, then carefully assembling and commenting the broad structure of the program. Subsequently, the process of implementing required features was smoother than expected. This is yet another miniaturization of real-world success – everything is easier if you communicate and plan before you act.

These two projects converge on what I would call the "issue" I care the most about in CS and beyond – mismanagement. It cannot be denied that every industry has a degree of wasted time and resources, but it is a tremendous and avoidable shame that much of it comes from poor planning and bad management. I believe that the UST CS curriculum tries to carefully prepare students for the reality of this problem, particularly through the Capstone course but more subtly through these different kinds of project structures. It is tremendously difficult at every level of work to manage and execute on a goal, no matter how simple it seems. Every step of a project's execution requires careful planning and strong communication, as well as independent drive and

a solid understanding of roles. This is where the final project of my portfolio comes in, a short essay on the idea of moral relativism.

This does not immediately appear to be relevant, but the core conceit of this philosophy is the communication of cultures, which is the most foundational aspect of communication. As we move from school into the workplace we will encounter many new people from a wide variety of backgrounds. The essential component of good workplace communication is in understanding one-another, and the impact that our personal lives and cultural upbringing has on our actual work, planning and execution is immense. If we want to do good work and avoid mismanagement at every level, we must always work towards common understanding. We cannot allow cultural or personal borders to entrap or undermine us – we must consider all perspectives in order to maximize the quality of our work and our enjoyment of it. That's one of the most important things college tries to prepare us for – how to interact with a huge group of people not-quite like yourself and work together to build the best possible system – to live in, to work in, and to thrive in.

That's what my portfolio is meant to represent – the analogous nature of programming to the broad realities of cooperation and progress. It is always my intent to be honest and straightforward about what is important to me, and if there's one thing that I hate it's miscommunication and mismanagement. I think that UST's faculty understand this well, and the courses provided clearly reflect that. As a result, the curriculum has expanded and improved my ability to combat these issues that I care about. These projects demonstrate the broad strokes of this process, and how it has affected me and my mentality. I hope that I can continue to apply these strategies and processes throughout my life beyond college.