3-29-2019

Woodland Cemetery CEL

Mickey Juranka
Western University, mjuranka@uwo.ca

Mickey Juranka

Dr. Barbara Bruce

29 April 2019

AH 4490F/G – 4493F/G

In this Final Report of my CEL Experience, I will give an overview of my work completed for the Woodland Cemetery, reflect on how the skills I've honed through SASAH affected my work, further reflect on the experience of creating and delivering my presentation, and discuss the impact the work may have on my life after I complete my undergraduate degree.

Overview of Work Completed

During my 1.0 credit CEL placement at Woodland Cemetery (henceforth "Woodland"), I coded a JavaScript application to be displayed on Woodland's website. The application was a project of my own conception, design, and execution. I used simple text editors (like Sublime Text) on my laptop and desktop computers to edit my JavaScript code and ran it through my Chrome browser to test the features as I implemented them.

My initial inspiration for the assignment was the plethora of art gallery apps I downloaded onto my iPhone while backpacking across Europe on my exchange. As someone travelling on a student budget, I couldn't always afford to pay extra for a tour of the famous galleries I visited. Thankfully, most of the large galleries have begun creating applications to give visitors access to recorded trivia and background details about the pieces they're interacting with, and I found myself wandering a single gallery room for an entire afternoon as I soaked in as much information as I could. After hearing my mentor lament the difficulty of actually finding specific gravestones in the cemetery, as well as the arduousness of conducting a tour himself, I had the idea to code a web

application that could be accessed by visitors in search of a more informative cemetery experience. The use would be triple-fold: freeing up some of my mentor's time and office hours, providing cemetery-goers with extra information and entertainment during their walks across the grounds, and giving visitors some directions and means of navigating the labyrinthine place of remembrance.

At the beginning of the development cycle, it became clear that I was the only one involved with the project that had prior development experience. I was an avid computer science student throughout high school and into my first year of university, and I consider it to be one of my deepest skillsets beyond those related to the arts education I've received with SASAH. Even though I didn't carry on with my computer science degree, I decided to maintain a flirtatious relationship with computer science work so that the avenue, as a career path, never closed for too long. Since I was the only one involved with my CEL that had the necessary experience, I had to create my own schedule and stick to it. The 4.5-month development cycle, beginning in earnest during mid-December and extending to the end of April, was organized into a list of goals that I sought to complete on a two-week schedule. Each of these goals would encompass a new learning curve and set of challenges. I will briefly discuss the most notable steps along the way of app development:

Visual Interface: After I mocked up what the interface should look like on paper, I began creating a visual representation with JavaScript. In terms of objective difficulty, is the most straight-forward step, since JavaScript is generally optimized for enhancing the visual appearance of sites and this is the area in which I have the most prior experience with the language. First, I created the map background by using a beige, flat background and loading a PNG file of the cemetery map overtop. I spent the next few days playing around with the placement of the dark

red map pin circles so that they could be reliably placed at their locations in the cemetery when the code called for it. However, since this was the first time, I'd picked up JavaScript in quite a while, I had to reorient myself through the tiring process of receiving errors, reading through your code to find your mistake, revising said mistake, encountering yet another error, and so forth.

Plot Database: Just as I began to wrap my head around my old JavaScript skills, I had to learn a whole new set of skills with MySQL, a database software that can help programmers create, store, and access data using their coding. While I began my using a text file as my information-storing document, I had to learn how to use this new software to store all of the information associated with the application in an elegant (and later editable) way, and how to get JavaScript to communicate properly with the database. After trying different methods, I decided to use the Node.js functions for interacting with the database inside of the JavaScript application. This step took me longer than I expected since I had a pair of new languages to learn, and this would be one of the responsibilities I'd offload to another developer first if I was working in a team.

Map Drawing: After the JavaScript program began to communicate properly with the database (which I verified by entering test phrases into the database and having the JavaScript program read and edit those phrases in my browser), I moved back to the JavaScript visuals. I spent a couple weeks learning how to optimize the screen redrawing (the term for the visual interface changing based on user actions) so that it didn't stutter or crash when the user scrolled around it very fast or clicked on too many things at once. Additionally, I spent a good amount of time learning how to create a path between the pins that I placed on the map (as illustrated in the pamphlet tours), only to later determine with my mentor that the feature should be removed. Sometimes in development, a week's worth of work can be deleted just like that. While having

more guidance and communication throughout would have minimized things like this, it served as a helpful reminder as to how the business functions.

Location Services: After I developed the window for displaying the gravesite information of a clicked map pin and imported some more of the pamphlet tour information, I set out to implement a feature that has since proven to be a massive headache. Essentially, I wanted to include a system within the app that asked users for their GPS location and displayed it as a moving pin on the map so that they could better orient themselves on the map. In an effort to get this application finished before my slew of final exams, I barrelled ahead with implementing the Location Services feature. In my own mind it had always felt like one of the most important features to include, since what initially prompted the idea for an application was the difficulty of navigating the cemetery grounds. Without a little moving icon to tell you where you were, how was an application any better than a paper pamphlet that prominently displays the exact same map? However, with little time to properly review the code I was typing in, I wrote whole sections of code on a mixture of self-assuredness and exam-fueled panic without properly testing it. After I had finished what I thought was adequate implementation, it all began to crumble very quickly, as every function I wrote returned error values instead of the correct system. Almost everything that can go wrong began to as my exams began to loom on the horizon; the application could communicate with some devices but not others, the application was struggling to connect to the online code library that contains key functions, the background map that the pin would be coordinated with either loaded on top of the application (instead of behind) and eventually caused the application to crash entirely. My last four weeks of work now felt like a virus infecting the entire product, since large portions were rewritten entirely to accommodate the new variables and functions required for GPS systems to work. In order to have the product deliverable-ready by the

end of April, I've decided to abandon the GPS services entirely, but that involves going through every section of code and removing all mentions of it so that the app can open without crashing.

Challenges during Development

There are some notable problems I encountered while developing that made it difficult to stick to my schedule. Firstly, I was excited to be getting back into my coding and my imagination was ignited with all of the different features I could implement. I was taking a mental tour through some sort of computer coder's candy shop and in the end, my mind's eye became larger than my skillset's stomach, by which I mean I was overly ambitious at the start of the project. I managed to develop most of the features I initially wanted, but at a much greater cost in terms of time and energy than I expected. Part of this discrepancy stems from the amount of time it has been since I worked with JavaScript; like any language, fluency in a coding language ages like milk without constant usage, and there's a good amount of online reading to be done in order to gain that knowledge and comfortability back. In the beginning of the process, it was incredibly frustrating and disheartening to see every line of code contain some sort of error, and when it's a project as individually-guided as this one, disheartenment can be the difference between a productive week and one of CEL avoidance.

While my breakdown of tasks into the above categories was useful for keeping myself on track, I found myself having to return and revise old sections of code often as things changed. My initial conceptions of how these systems would run were completely changed upon trying to integrate both the MySQL database and the GPS services, and I had to completely re-write sections that I wasn't expecting, which in turn caused me to revise every section that interacted with those sections to ensure everything ran smoothly, and it was a huge headache.

One of the biggest tasks was simply the amount of learning I had to do to pull this off, and with no direction as to where to begin looking and whether I was ever on the right track. While this ended up being tolerable due to the sheer amount of resources that are available online for free, the entire process felt overwhelming at times. I definitely felt at times that I was single-handily developing an application that a whole team should be developing, and with only half the available time. While my desire to finish each goal on schedule kept my pace of learning and feature implementation brisk, it also led to some deeper issues with my code that another pair of eyes, or an experienced supervisor, may have identified earlier and saved me the trouble of what felt like ripping out sections of code by the roots.

Mentor Relationship and Application of SASAH Skills

I only had a few meetings-worth of communication with my mentor over the four and a half months of development. He was still getting oriented in his position at Woodland, and most of the time I felt I didn't need any information from him, and so neither of us reached out often. While I don't think the lack of communication my project was negatively impacted in a large way, there are nonetheless areas in which my time could have garnered more value if we'd been had more contact.

Firstly, as I touched on briefly in the "Map Drawing" section above, some features became redundant or pointless retroactively. A more open and frequent line of communication would have let me bounce ideas off him to adjust both of our expectations and make sure I wasn't wasting my time. Additionally, showing him some of my expected steps in my original plan (such as the 'Website Optimization' step, in which I planned to access the Woodland website and see how my file ran on it) would have alerted me to peculiarities in the process, such as Woodland not having regular access to their site, a fact which forced me to arduously change my notation style to be more accommodating to whoever has to complete that task in my stead.

Secondly, I feel as though I could have learned more about what Woodland offers the community beyond the tour aspect and incorporated that into my project. While most of my inspiration for the look and feel of my application came from art gallery applications, there is something about the particular role a cemetery plays in our society that minimizes the location as a whole by playing up the entertainment value (as if it were a tourist attraction). That being said, value could be attributed to getting people to the cemetery in general, and perhaps destigmatizing the area in an effort to make it more of a historical site, as opposed to just a place of mourning. Obviously, there are a lot of conversations that could have been had about this topic and I regret that most of my CEL consisted primarily of replicating the features of other applications, as opposed to exploring what could set an application for Woodland apart from the others.

However, something of particular value that came out of being paired with a non-coding mentor is the vocabulary I had to develop in order to communicate computer related concepts to someone who was previously unfamiliar. It didn't behoove me to teach him everything from the ground up for the purposes of communicating what I was doing, and so it became

If there is a "community" element to my Community Engaged Learning experience, I would say it is the function of the application itself. From looking at the themes of the tours themselves (which cover topics like London's historical art community, famous women, wacky stories from the time of Confederation, etc.), one can see that Woodland is trying to encourage the London residents to visit and engage in the history of London. After all, a tour of Woodland could be considered a historical tour of London's past residents, and the information about each interesting character can help visitors marvel and reflect about the amount of history embedded in this city.

When it comes to the skills I applied to my work process, SASAH had a clear role in influencing my approach and work ethic throughout the project. One of the things SASAH does best is giving students room to approach topics from their own point of views and skillsets, which leads to the variety of final pieces we produce, including visual artwork, essays, musical compositions, and short stories. With my CEL project, I felt empowered to suggest an approach that I believed gave me a chance to offer a unique service for Woodland, as opposed to being quietly relegated to a duty that may have otherwise given them less value. Additionally, SASAH courses generally encourage additional student initiative and self-reliance, and I was able to use those both in planning my development cycle and holding to my self-imposed deadlines.

On the other hand, there were also challenges to working on a computer science project alongside my final year of primarily Arts-focused coursework. Halfway through the development cycle I reflected on the impact I believed the assignment was having on my regular work ethic. Coding relies on a different mindset than research and writing, and the more I spent on one, the harder it became to switch to another. While I code, my brain is focused on repetitive, incremental, and methodical tasks and my mental rewards system becomes trained accordingly; with every small portion of coding being as important as the next to the overall functioning of the program, I became hooked on the short-term gratification of solving small problems over and over. On the other hand, the longer-form process of researching a topic and writing a written response or analysis gave less in the way of gratification. Even after my papers were completed, I wouldn't have any indication as to whether I'd performed adequately until weeks (or even months) later. As such, in hindsight I would have preferred to work on a project like this during the summer so that it didn't have to compete with wildly different (and yet similarly-intensive) tasks.

Presentation Experience

Creating the presentation was a fairly straight-forward experience for me, but I prepared for it in a rather unconventional manner, at least as far as my own presentation-preparing strategies go. As I sat down to create my PowerPoint presentation, I remembered a particular online video presentation whose presentation style really resonated with me. After digging through my YouTube history, I finally found it: "Magic the Gathering: Twenty Years, Twenty Lessons Learned", a filmed presentation from the 2016 Game Developers Conference in which a lead designer covers twenty things he's learned about game design. The content in this case mattered to me less than the presentation style. The speaker would open with an initial hook, like a funny image out of context, give a similarly humorous anecdote before leading into the "point", before showing the point spelled out on screen and a repeated reading. Something about the final reiteration, instead of simply wrapping up the section verbally and moving onto the next bullet point, felt like a strong way to let the presentation sink into the mind of the viewer.

It may be easier to explain my presentation strategy with an example. From the section "Challenges", I would first show an image of a women sitting down in front of a computer, holding her glasses in one hand and pinching the bridge of her nose with the other. This image is my initial hook, which sparks a bit of laughter as the audience immediately gains some insight into what's coming next, and rather than engage directly with the image, I begin discussing the problems that come with getting back into a coding language and the initial difficulties I encountered. When my point and explanation are completed, I finish my showing the thought in a succinct manner (in this case: "coding is hard and I did it a long time ago") and read it aloud to the audience before I move onto my next hook. I'm incredibly excited to use this format and style again in future presentations. Rather than plainly listing the bullet points and elaborating on each one in turn, this

process of "hook, elaboration, and point reiterated" feels more entertaining to both present and watch.

I felt it was important to include a good amount of humour in my presentation since the subject matter of computer coding is foreign to many of my fellow SASAH members. The actual computer terms would be meaningless jargon to most of them, and many of the challenges that come with coding could only be crudely explained through allegory. As such, I felt that discussing my emotional experiences throughout the process in a funny way would be the most interesting thing for them to hear about, since many of them would have had to undertake similar projects in similar circumstances before (maybe even for this project). At the same time, I also wanted to coax them into a comfortability with discussing computer coding; with a project like this, I imagine a presentation gone wrong would result in a bored or bewildered silence, and I really wanted to show the humanness behind computer coding. While the numbers and algorithms involved in a development project are very cold and calculating, there is an important human element that's crucial to making something like this work; it includes the leadership involved, the unconventional problem solving, the resilience to persevere through exhausting and frustrating work, and the communication skills needed throughout development. While I don't think many of my SASAH classmates are going to pursue a career in coding, they may have to work on a development project like this in some way, and learning about the challenges I faced might help them to more easily anticipate and support their co-workers while they're conducting similar work.

Skill Development and Impact on Future Career

When I graduated from high school, I was comfortably convinced that I would pursue a career in coding projects after undergraduate studies and pursuing a double-major through SASAH was an attempt to incorporate my Arts interests alongside the crunchier math content of a computer

science major. However, I immediately found my computer science courses to be unsatisfying; I'd already the content of the first two years of an undergraduate degree in my high school courses. This, coupled with often hearing about how experience is more valued in the field than formal education, led to me devoting more of my studies to my non-computer science interests such as Linguistics and then Political Science instead of a computer science major.

While I've worked on coding-related things during my summers, this project gave me both the chance to dive headfirst into a true coding challenge and, as opposed to a rushed 'code-a-thon', the time to develop my skills accordingly so that I could make the most of the opportunity. Using computer code is in my ways akin to a language, and with disuse you forget the small things that separate a fluent speaker from someone who took a few electives; how to phrase and order your statements to achieve the meaning and effect you intend is the most important aspect, but beyond that lies the short-hands and tricks that allow your program to function faster and slicker, and, in something fairly unlike regular language, how your language fits in and can be used alongside other languages. This project allowed me room to explore the more advanced side of programming, with combining my language with other systems (such as MySQL databases) and expand my ability to use JavaScript; while I was previously comfortable with using JavaScript to add animations and other aesthetic touches to a website, I'd never attempted to code a full application inside it before and using it to code the background functions forced me to learn a new set of functions and strategies for organizing information within JavaScript.

As I anticipate the 'career path' of mine that lies in wait beyond my upcoming graduation, I'm glad I'll have this CEL experience to consider when applying for computer science jobs. I expect to further develop this skillset over the next year as I make some money and apply for political science graduate programs and it'll be nice to inform potential employers about all the

new skills I acquired through the CEL and perhaps even show them the application. Beyond this next year, I'm still unsure of the role coding as a whole will play into my life. It's a stable career path with a lot of foreseeable employment opportunities, and to boot it's a constantly changing and developing field that won't stop throwing new challenges at you to sink your teeth into… and yet, I can't help but feel my heart sink to imagine myself still clickity-clacking away at a keyboard in five or ten years' time. This CEL reaffirmed so much of what I both loved and didn't about computer science; the methodical, incremental approach to working on a project is very much in line with my own thought process, and the small, tangible goals really fuel my motivation to keep working. On the other hand, it can feel incredibly isolating and disconnected from the world around me; something I enjoy about political science is the immediate relevance of my studies to the world around me, whereas computer language and all the constructed forms it can take really are a separate world unto themselves and it isn't a world I'm sure I want to travel to often, let alone permanently relocate to. Regardless of whether my future lies in coding or not, this project as a whole was an interesting and engaging experience and I'm glad that I was able to cap off my graduating year with what became a varied and challenging undertaking that reflected, as so many SASAH projects do, my own academic journey and interests.