Matt Walsh

W0075056@nscc.ca

INFT4000 Special Topics

Final Project Reflection

Contents

[Portfolio Integration 2](#_Toc58615064)

[My Role 2](#_Toc58615065)

[Search 2](#_Toc58615066)

[Hidden Material Box 2](#_Toc58615067)

[Technology 2](#_Toc58615068)

[Reflection 3](#_Toc58615069)

# Portfolio Integration

This document is a portfolio piece titled “[Fizz Nizzard’s Spell Book](https://github.com/matt-walsh/fizznizzards-spellbook)” located on the Github account of [Matt Walsh](https://github.com/matt-walsh/), the author.

## My Role

My role in this project was all encompassing. I was the project manager, designer, and the designer. As the designer, I used my knowledge of design to create an interface that was intuitive for the user. As the developer, I used my knowledge of HTML, CSS and Javascript to create the design, and implement the functionality. As the project manager, I had to decide how the project was going to be completed, and if features would have to be cut due to time constraint.

Unfortunately, two features had to be dropped on the final day due to time constraints.

### Search

The search feature had to be scrapped, due to the implementation of the spell filter. It is currently planned for version 2.

### Hidden Material Box

The Spell Materials box is supposed to be hidden when the user has not selected the materials check box. This was cut due to issues that occurred in the CSS when the box was missing. It is currently planned for version 2.

While it was unfortunate to have to cut these features from the initial release, I look forward to completing them in version 2.

## Technology

I used several technologies to create this application, but the two main ones are:

[React](https://reactjs.org/) is a web framework by Facebook, which allows developers to create component-based applications for the web.

[Electron](https://www.electron.org) is a web framework maintained by the OpenJs Foundation, which allows developers to create native applications using web technologies.

I would classify myself as intermediate in both technologies, however, I found that combining them was quite difficult. I ran into several problems, such as issues with Electron refusing to render a react application.

## Reflection

during this project, I discovered a lot about both how Electron, such as how it sends events(message) from the main process to the render process. As well as about react, like how component lifecycles within react run and when (what triggers them).

I think my most significant discovery was how little I know about the Node runtime. I realised it was a powerful tool when, during my initial setup phase, I ran into the problem where Electron and the React server would try to start at the same time. Because of this, When Electron started first, React would complain about missing data, as Electron had already sent the data to react, which wasn't ready. When React started first, everything was fine.

My initial solution was to use `npm start`, wait for the React server to start, and then start Electron (`npm run electron`). I quickly decided that there had to be a better way. and I began investigating npm packages related to automation, and I discovered [Foreman](https://www.npmjs.com/package/foreman) which is a npm implementation of a procfile manager written in Ruby. A procfile is a file containing commands to be run on a cloud server but works just as well on a home pc. Using this, I was able to then write a small Javascript file (launchDev.js) which waited for a connection from the React server, and then started Electron. This solution was not perfect, but it allowed me to continue my applications development.

(React server will still open a browser tab with the React running inside. However, due to requiring a data connection to function, the app just crashes.)

This issue allowed me to explore the capabilities of the Node runtime. It was one of many that allowed me to understand the underlying power of many of our tools. Working with and around issues, for me, is the best way of learning. I believe, now, I have gained a slight glimpse into what is possible using Node and Javascript.