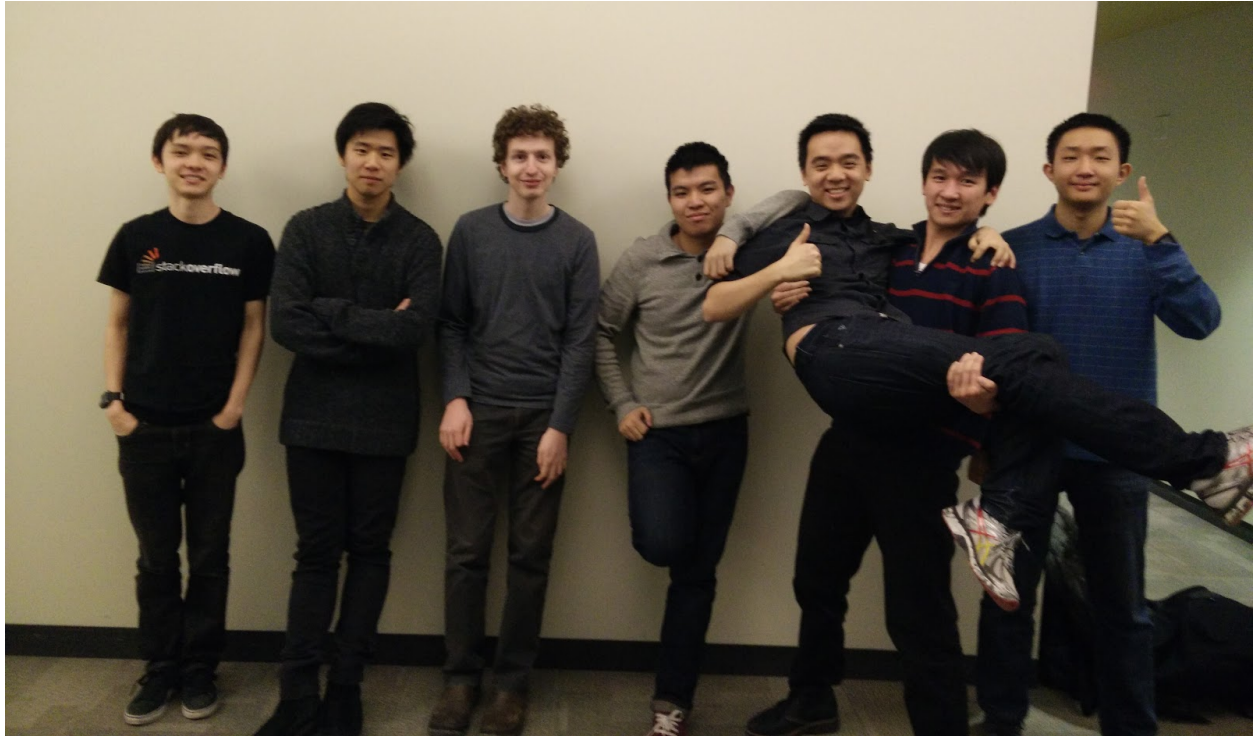# Group 4 - Aquarium

Team Introduction

Spencer Elliott

A passionate software engineer who is always striving to learn new concepts and keep updated with the latest technology. An avid supporter and contributor to open-source projects, he always aims to make collaboration easier through project structure and documentation. He has industry experience in building and deploying web applications, using technologies such as AngularJS, Ruby on Rails, and Docker.

WeiPing (Allen) Hsiao

A software developer who loves to use the power of programming to make everyday life easier. Stays ahead of the curve by keeping up to date with backend technologies, including Ruby on Rails and .NET. Previously interned at a startup company and focused on developing client and server interactions.

Alexander Biggs

A game developer with industrial experience. Enjoys managing the creative aspects of the project and ensuring that the team has a consistent vision. Adapts to new technologies quickly and works well under pressure. Helps run the Game Design and Development Club at the university.

Peter Lu

Hi, I'm a third year Computer Science student current in the Specialist stream at the University of Toronto, who's goal is to make learning accessible and natural for everyone. I really enjoy helping others, and work best in groups, where I can discuss with people about all aspects of the project. On my spare time, I enjoy sketching and reading novels, and like to keep with the most cutting edge technologies that would help make people's lives easier.

Nicholas Goh

Hey, I'm a third year Computer Science Specialist who enjoys nothing more than making cool and interesting things with a cool and interesting group of people. I enjoy going to events such as hackathons, simply to enjoy the innovative atmosphere and have attended several in the past two years. In addition to your regular computer science hobbies, I enjoy music of all kinds, and reading whatever catches my eye. I'm excited to be a part of this group, and look forward to working hard to make the best product that we possibly can.

Ryan Fan

Third year computer science specialist, problem solver and critical thinker. I consider myself a good listener who can provide honest opinions to the team. Things I enjoy doing are playing trading card games, surfing reddit, playing pool and going to the gym. I hope to be an active contributor to our group and can not wait to see our finished product!

Kyle Zhou

Hey, I'm a second year Computer Science Specialist who loves to code and to make cool things. I enjoy working on projects with a group, which is why I enjoy attending events such as Hackathons. Some of my hobbies include skiing, browsing sites such as Reddit, and tinkering with my devices and gadgets. I am excited to be a part of this project, and I hope to help make the most beneficial product we can.

Choosing The Project

Our group decided to develop an emulated terminal named Aquarium that would exist as a learning tool, and a pseudo-integrated development environment (IDE) for both new and experienced shell users.The naming Aquarium comes from our intent for it to be a complementary wrapper to the learning shell - FISH. Aquarium would be equipped with several tools inspired by modern IDEs such as inline and searchable documentation, as well as a variety of auto-completion and flag suggestion settings which are provided as optional settings. The group decided to pursue this topic unanimously as we felt that this idea was the most appropriately sized for the scope of the course, while still providing a challenge in production. In addition to this, this project idea showed the most potential utility in terms of target audience size and representation, and should our interest in the project persist, this project has lots of room for expansion during continued development.

Personae

Keen Kevin is an eager young computer science student going into his second-year who just happens to be free during his summer. Kevin is really excited for his next year at the University of Kansas and wants to get ahead early and learn how to use one of the important tools for his course, the computer's Shell system. Kevin unfortunately doesn't have many friends that are well-versed in computers, and has already read several online resources related to the subject, but had difficulties following along, especially with no one to ask for help.

Novice Nancy is a student currently taking third-year computer science courses, but hasn't had to use shell in quite a long time. Nancy is not actually new to using Shell and has already used it previously in courses that required basic knowledge such as downloading, removing, changing directories, etc. However, Nancy has unfortunately forgotten most of her knowledge after not using it, and wishes to review. Nancy has friends she could ask to teach her, but she does not want to out of embarrassment, and does not want to read a tutorial as she thinks that she can recall how to use it better as she uses a shell more.

Techy Timmy is an avid tech-enthusiast with lots of computer science experience who enjoys nothing more than experimenting and toying with the latest technology. To Timmy, there is nothing better than the feeling of improving his experience with a difficult task or an interface. Timmy already knows how to use a Shell perfectly and even chooses to use zsh over bash for customization purposes. Timmy is interested in seeing what new technologies can improve the somewhat archaic and unintuitive interface of a shell that he has grown to love and hate.

User Stories

As Kevin, I want to learn Shell in a beginner-friendly environment despite the fact that I am not familiar with technical terms such as directories and permissions.

As Kevin, I would like to learn Shell in a visual fashion to compliment already existing text-based tutorials.

As Nancy, I would like to review the core features of a Shell without having to deal with extremly basic tutorials.

As Nancy, I would like to develop my skills with the Shell at my own pace with the help of suggestions or reminders from the system of syntax or format.

As Timmy, I want a completely customizable experience ranging from an entirely unobstructed experience to a full-blown tutorial.

As Timmy, I want this shell to be no less usable than any other shell that I use such as bash or fish.

As Timmy, I want access to a tool that I can use to both improve my experience with Shell, but also to improve efficiency of any of the tasks that I perform in Shell.

Minimum Viable Product

At the end of our 9-week period, our minimum viable product will be a desktop application that resembles our mockup below. This program will have access to the user's built-in shell just like any other terminal emulator. It is meant to be an extension and possibly be a replacement to the most basic terminal window that lacks support for inexperienced users.

Our mockup currently consists of a terminal window and a side panel. The terminal will just like any other terminal found on most computers, black background and coloured text, no images. The side panel is what makes our program unique and helpful to our users. The side panel responds to user input by looking up documentation and producing logical amounts of information to display.

Since users come from different levels of experience, attention needs to be paid at how much information needs to be shown to the user. Users might be able to configure how much information they wish to display, depending on their experience level and preference. Another choice we can do is to have documentation displayed with a show and hide button, but this requires additional clicking from the user.

Overall, we want our application to be a terminal that is designed with the beginner programmer in mind, but designed smartly such that it doesn't hinder a more experienced user. Our application will provide not only the most basic functionalities of any system shell, but many extra features that we have mentioned such as in-line completion, searchable documentation, and ideally much more that can aid experienced users, innovating and improving on the time-tested tool that programmers love and hate.

Release and Iteration Planning

For our first release, the following will be implemented:
- As Timmy, I want this shell to be no less usable than any other shell that I use such as bash or fish.
  - Basic terminal emulator window, similar to Terminal.app on OS X
- Built binaries for OS X and Linux, interfacing with fish shell

Postponed until next release:
- Side panel with searchable documentation: since this can be implemented after the terminal emulator itself, we can distribute development time more evenly by implementing this later.
- Interactive tutorials: (same reason as above)
- As Nancy, I would like to develop my skills with the Shell at my own pace with the help of suggestions or reminders from the system of syntax or format.
  - This user story asks for an auto-complete or auto-suggest feature which is within the scope of our project, but is postponed as we are prioritizing the core features and functionality first with additional tools such as this to be postponed.

CRC Cards
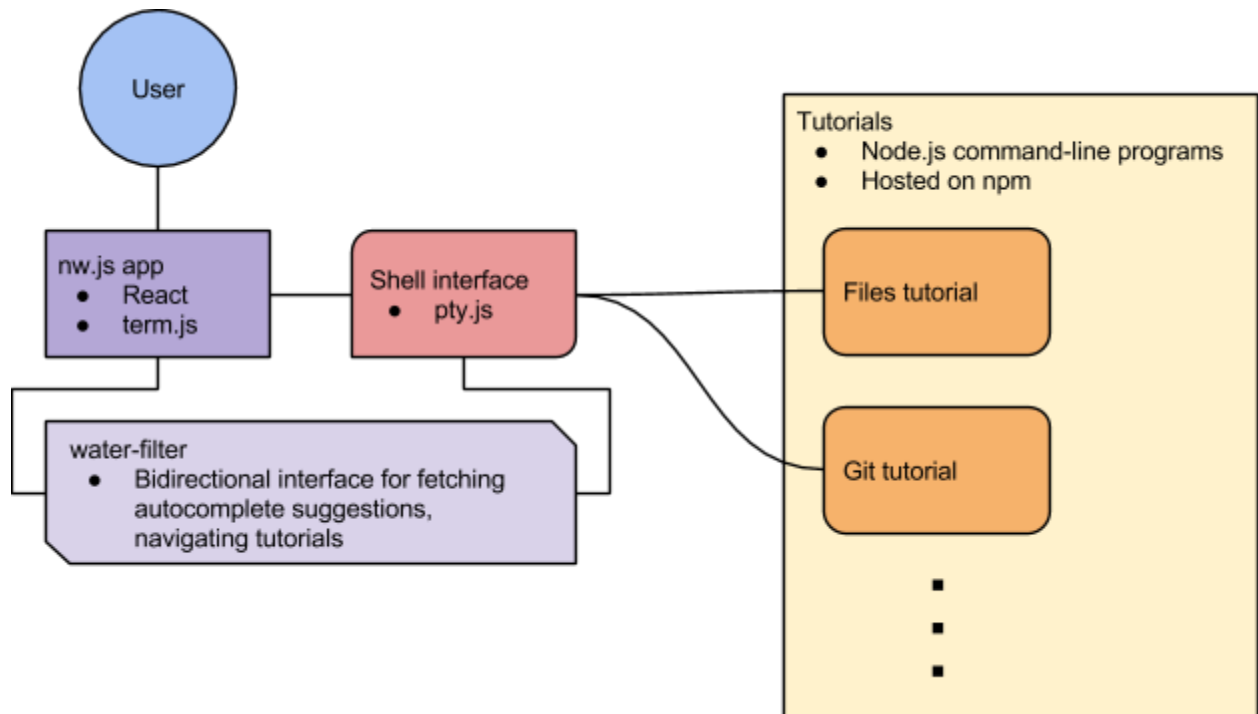
Terminal	
• Emulation of a native shell to receive user input • Displays output from the shell	• Shell • wFilter

Shell	
• Provides two-way communication between the Terminal the OS • Allows for forking of new Terminal processes	• Terminal • wFilter • Tutorial

wFilter	
• Augments the default terminal to support additional functionality • Provides features such as syntax highlighting, autocompletion	• Terminal • Shell

Tutorial	
• Holds all the content of a user-created tutorial	• Shell

As Kevin, I would like to learn Shell in a visual fashion to compliment already existing text-based tutorials.

- Using Aquarium, Kevin would primarily be interfacing with the Terminal class, which is a representation of a normal shell, and as he is typing performing commands, most of the visuals that would be expected from a visual learning tool would be brought from methods in the wFilter class, giving Kevin access to visual features such as multi-coloured syntax highlighting, and in-line documentation that would compliment his user experience.

As Nancy, I would like to review the core features of a Shell without having to deal with extremely basic tutorials.

- Using Aquarium, Nancy would have the option to be able to skip the initial tutorial designed for beginners, implemented as part of the Tutorial class, and instead have a faster review of core principles from reading inline documentation provided by the wFilter class.

As Timmy, I want a completely customizable experience ranging from an entirely unobstructed experience to a full-blown tutorial.

- Using Aquarium, Timmy could easily customize his shell in any fashion that he wishes, either browsing and working through the various tutorials as part of the Tutorial class, changing settings and enabling or disabling features provided by the wFilter class, or simply just use Aquarium as a normal shell with the Terminal class.
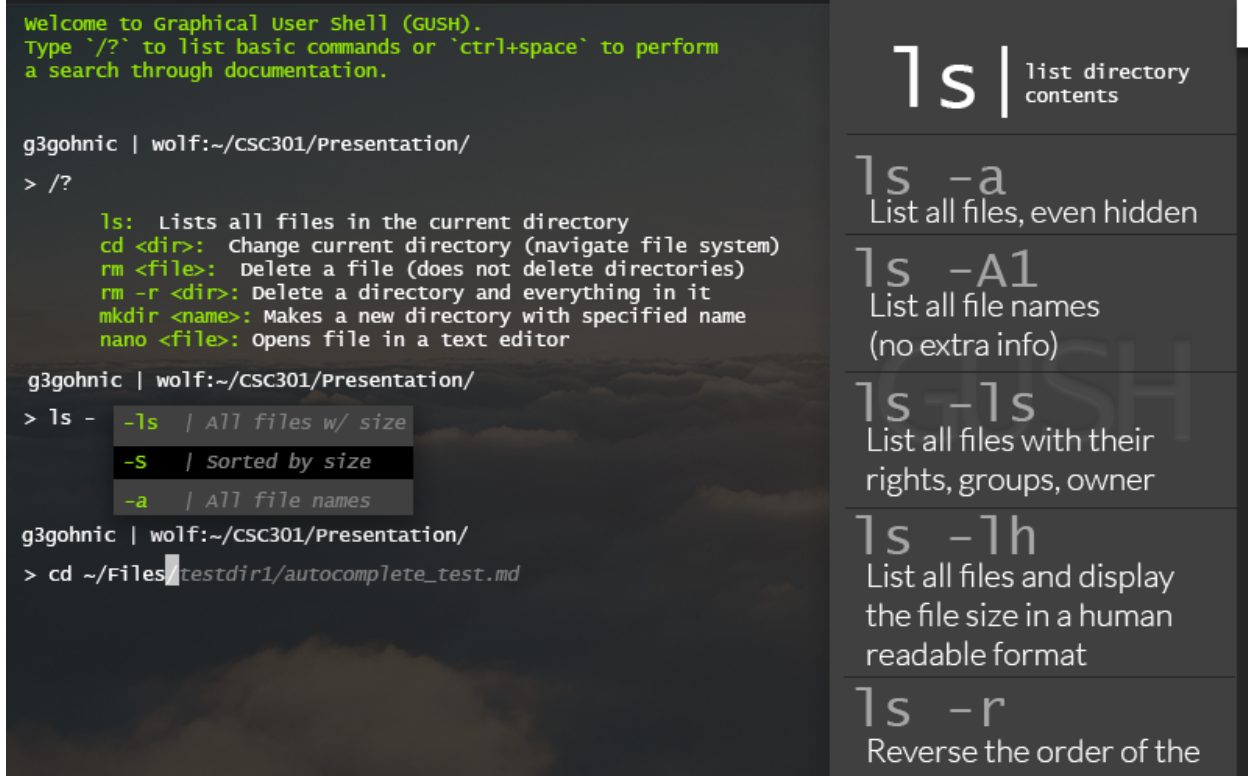
Appendix

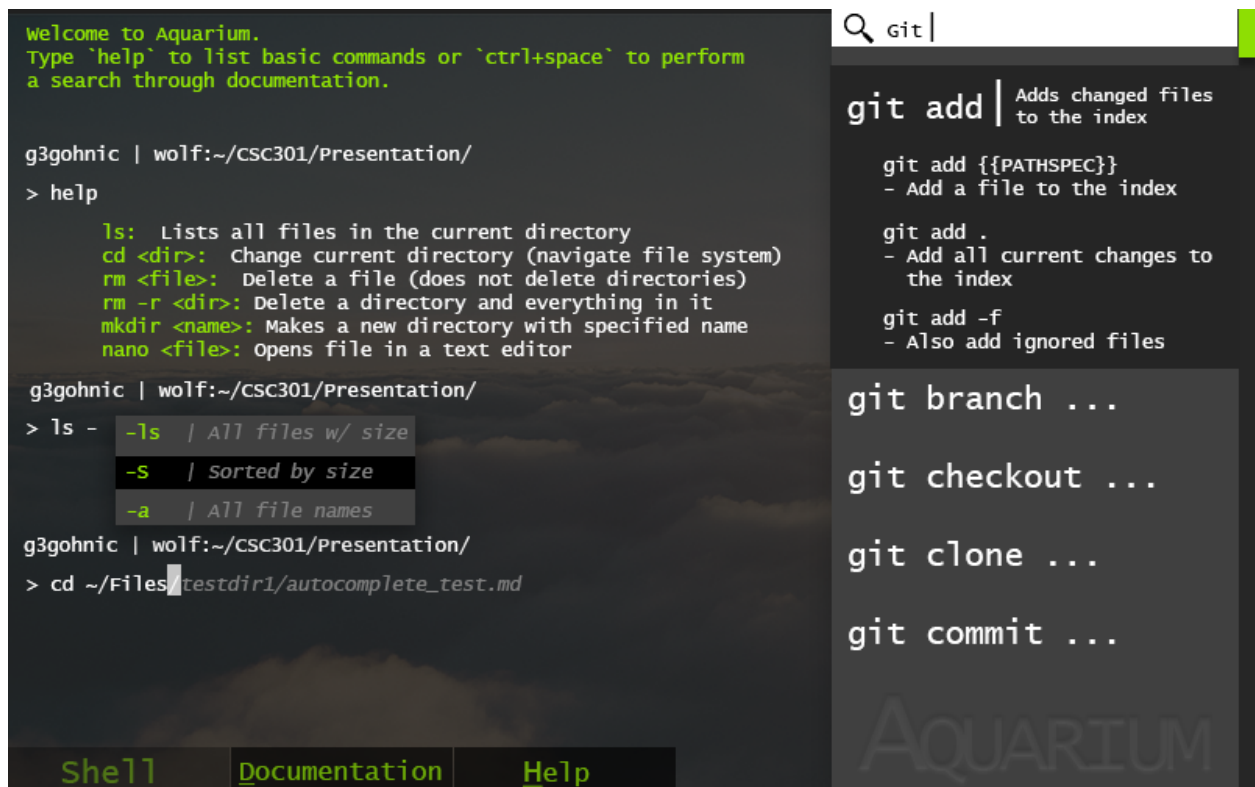Figure 1 - Initial mockup presented during pitches

Figure 2 - Mockup after several revisions based on feedback from initial mockup

What has your experience using a Shell been like thus far?

What is the most common operation you use the Shell for? (describe your common workflow when using shell)

What do you consider to be the most important feature of the Shell (or don't know)?

What do you think was the hardest thing you've had to learn/understand so far?

Would this be easier taught through the use of diagrams?

What do you think is the most intuitive part of the Shell?

What do you think is the least intuitive part of the Shell?

What do you think is something that all Shell beginners should know?

Have you used any pre-existing shell tutorial resources, if so, which?

How have you gotten over any issues that you had with using the Shell?

Have you tried using any command line editor such as nano/pico, vim or emac? If so please briefly describe the experience.

Figure 3 - User studies questions used