British Columbia Instiute of TEchnology

***Computer Systems Technology***

Comp1111 – Essential Skills for Computing Final

Lab Instructor NAME: Nikhil bhardwaj

Set: 1B

Title:

Atom Code Editor

Team Members:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Student Name** | **Student Id** | **Email address** |
| **Student 1** | Albert Chen | A00961560 | [Albert.chen.97.00@gmail.com](mailto:Albert.chen.97.00@gmail.com) |
| **Student 2** | Mark Ennis | A00941761 | [mark.andrew.ennis@gmail.com](mailto:mark.andrew.ennis@gmail.com) |
| **Student 3** | Andy Lin | A00954644 | [Andylin1218@gmail.com](mailto:Andylin1218@gmail.com) |
| **Student 4** | Syd Sisco | A00943902 | [sydsisco@gmail.com](mailto:sydsisco@gmail.com) |

**COMP 1111 Final Presentation: Atom Code Editor**

**Introduction**

Atom Code Editor is a completely free, open source code editor developed by the GitHub team. It was designed to be customizable to the core, while retaining a focus on user-friendliness.

**Why should you use Atom?**

Due to the fact that Atom is completely open-source, it is highly extendable. There are a wide range of pre-built packages available for no cost. Many of the more popular packages are integrated into the core download, resulting in a product that evolves with customer use. If you are not happy with one of the included packages, you can go into your package manager to disable or uninstall the offending package. We will demo some of the optional packages that we found to be interesting and useful.

You can also build your own packages. If you want to add a new feature, build one yourself! It will look good in your portfolio.

Atom also has some interesting features that link it to GitHub. When working from a repository, the files in your working project folder will color code in conjunction with the differences to the repository you are working from. There is also a package that allows you to push and pull changes directly from your repository, from Atom.

**Definitions and Principles**

**Atom Features**

Atom comes with several built in features, some of which are unique to Atom, at least in our experience.

**GitHub Repository Color Coding**

Thanks to the fact that Atom was developed by the GitHub team, there is a strong link between Atom and GitHub. When you pull files from a repository and open them in Atom, there is a built in color coding system that Atom uses to show you which files you have made changes to since you have pushed or pulled from your repository. Green indicates that the folder or file is something new that has been added, and yellow indicates that a change has been made to this file. Not only that, but if you are working on files, and then you pull or merge changes from your repository, Atom will automatically update your working folder and files.

**Snippets**

Snippets, while not unique to Atom, are a highly useful tool for developers. You can easily create your own shortcuts that will allow you to quickly generate code that repeatedly use. Below is an example where the syntax is explained:

First go to *File > Open Your Snippets*

Your snippets.cson file will open.

'.text.html.basic':

'insert external css link':

'prefix': 'css'

'body': '<link rel="stylesheet" type="text/css" href="${3:style/${2:style}.css}" />$1'

The first line establishes what language you are writing in, in what kind of file: *'.text.html.basic':*

It is important to note that if you have multiple snippets that fall under this same heading, do not repeat this line. Just type the new snippets under the first.

The next line provides a brief description of the snippet you have written: *'insert external css link':*

The *‘prefix’:* is what you type out to auto-generate your code.

The *‘body’:* is the auto-generated content. This example also demonstrates the tab function of snippets. The dollar sign you see in the body, along with the curly braces, show where your cursor will end up after the snippet auto-completes. In this example, your cursor will end up at the end of the snippet, at the *$1*. This is convenient is you are consistently naming your styling sheets style.css, and placing them in a folder called style. If however you have a CSS file that isn’t name style, hit tab once again. The cursor will go to the second dollar sign (*${2:style}.*). The dollar sign, followed by the braces, will select anything inside the braces. If you were to hit tab again, the entire file path would be selected, allowing you to easily modify the link to your style sheets with only the push of a few buttons.

**Multiple Cursors**

A really useful feature of Atom that is easy to use is the Multiple Cursor. All you need to do is Ctrl – Click somewhere in your document, and another cursor will appear, allowing you to type the same thing in multiple places. This feature can be quite useful when you need to add a new class to multiple elements.

**Packages**

Atom calls all of its plug-ins packages. They are developed by the Atom user base, and made available on GitHub. You can also browse for packages in Atom. Here is a list of the packages we will demo:

**Mini-map**

Mini-map provides a small, scroll down window on the right hand side of the working screen. It allows you to see where you are in your code, and makes it easy to sections of your code when you are working in larger files.

**HTML Live Preview**

HTML Live Preview allows you to view the changes you are making to your website as you type them in. Similar to the live preview function in Brackets.

**Sublime Vertical Select**

Taking a page from the Sublime code editor, a user that created a vertical column select. Holding down Alt, and dragging your cursor down the page, will allow you to select columns of the page. This allows you to easily modify indentation in your code.

**Git Control**

Git Control allows you to use some of the more commonly used git commands directly from Atom while working on a repository. You can create, switch or checkout any branch. You can push either the entire working folder, or just a selection. You can pull from Atom. You can reset any file to its previous state with checkout. All of your commands will be logged and visible.

**Pigments**

Pigments will scan your source files for colors, and then create a palate to display these colors. It updates the colors as you make changes. Pigments is highly useful for design purposes.

**Challenges**

The most challenging aspect of Atom is discovering some of the finer workings of using Atom with GitHub. Git Control required us to create a SSH key for our personal laptops, as opposed to one from a Git Hub client.

**Conclusion**

Atom is an easy to use, customizable and constantly evolving code editor. It has been built and expanded upon by its users, and allows contributors to add their favorite parts of other editors to Atom. The GitHub integration makes Atom ideal for developers working with teams or versioning.

**Distribution of work**

We attempted to share as much of the workload as possible, allowing each member of the team to learn as much about Atom and share it with each other. This is especially true of the research that we did. That being said, the website was largely put together through the efforts of Andy, Albert and Syd. The majority of the word document was written by Mark.

**References:**

Atom Homepage:

<https://atom.io/>

Article Promoting Atom:

<http://www.hongkiat.com/blog/atom-code-editor/>

Various Atom Packages on git-hub:

Sublime Vertical Select <https://atom.io/packages/Sublime-Style-Column-Selection>

Mini-map <https://atom.io/packages/git-control>

Git-Control <https://atom.io/packages/git-control>

Pigments <https://github.com/abe33/atom-pigments>

Html Live preview <https://atom.io/packages/atom-html-preview>