CSE 3150 - Basic Text Editor Manual

By Matthew Rumbel

Overview:

This project will allow me to practice C++ programming by writing a console application: a vi-like text editor. Throughout this project, I will learn and gain experience with using different object-oriented approaches, such as Model View Controller and Observer pattern styles of design. Although the text editor itself is not particularly exciting to implement, the point of this project is to gain experience implementing a non-trivial program following the OO paradigm.

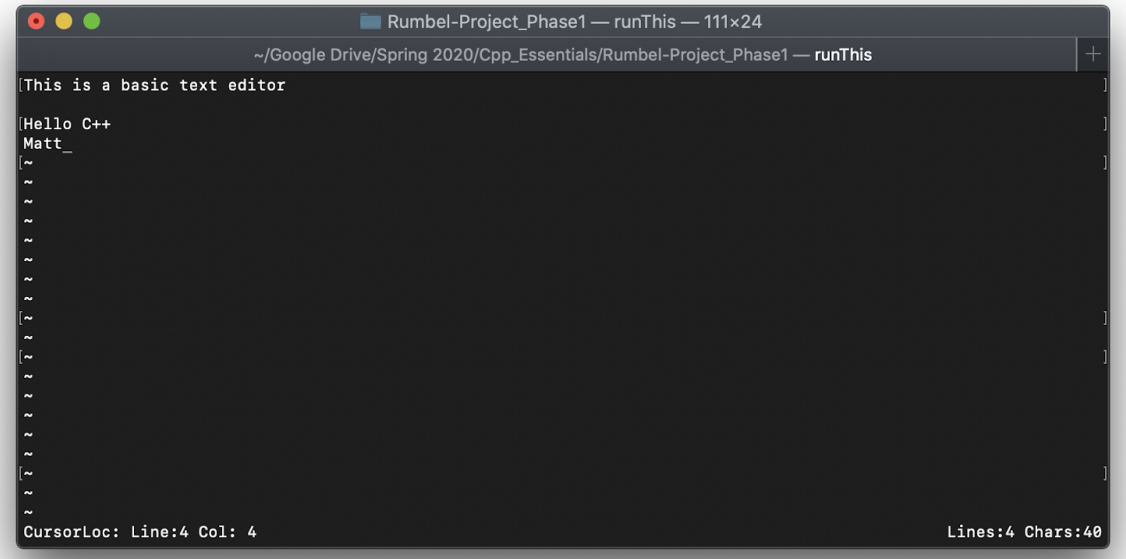
Table of Contents

Compile and Run

To compile this and run this program, please use the *make* utility on a Unix-based system. Using the included Makefile, the *make* utility will compile the necessary files and create an executable titled 'runThis'. A user can run the command **./runThis** to start the program.

```
$ cd Rumbel-Project_Phase1
$ make clean
$ make
g++ -c Commands.cpp -o Commands.o -O3 -std=c++14 -Wall -I.
g++ -c Document.cpp -o Document.o -O3 -std=c++14 -Wall -I.
g++ -c ECEditorTest.cpp -o ECEditorTest.o -O3 -std=c++14 -Wall -I.
g++ -c ECTextViewImp.cpp -o ECTextViewImp.o -O3 -std=c++14 -Wall -I.
g++ -c EditorController.cpp -o EditorController.o -O3 -std=c++14 -Wall -I.
g++ -o runThis Commands.o Document.o ECEditorTest.o ECTextViewImp.o EditorController.o
$ ./runThis
```

After the last command is run, the text editor will open and will be ready for text manipulation.



Features

In this current version of text editor, functionality for basic text editing features are available for use. These include, inserting characters, carriage return, undo, and several others. Here is a list of all available features:

- Text Insertion
    - A /symbol can be inserted into the editor by pressing a desired character key from the keyboard (for example, pressing the letter **a**, number **7**, or symbol **+**.
- Text Deletion
    - A piece of text can be removed from the editor by pressing the **backspace** key (or **delete** key on MacOS). This will remove the symbol at the cursor position, if there is one to remove. If the cursor is all the way to the left of a row of text, the text deletion feature will move the text on the current row to the end of the previous row (if on row > 0).
- Line Break
    - A user can insert a line break by moving the cursor to the desired location and then tapping the **enter** key (or **return** on MacOS). On the same line, if there is text after the cursor location, that text will be moved to the new row.
- Undo/Redo
    - In this implementation, Undo and Redo will behave in a similar manner as other common text editors. As a note, you cannot Redo a command (text insertion, deletion, etc…) that has not been Undone. A user can press **CTRL-Z** to Undo and **CTRL-Y** to Redo.
- Quit
    - A user can quit out of the text editor by pressing **control** and **Q** (**Ctrl + Q**) keys at the same time. This will terminate the program.

User Interface

In this implementation, once text insertion has begun, the user will see a status bar located at the bottom of the text editor. The status bar will show useful information such as total lines and total characters within the editor, as well as current location.