Frances Vu

Dr. Rivas

Software Development I

8 May 2017

Type It

Abstract

This document contains how *Type It*, a java programmed application, works within its current state. This includes a detailed system description which specifies what the application does, how specific users interact with the application, and the physical requirement of system resources. As there are similar applications already available, these other applications are briefly discussed along with how *Type It* differs from these other systems. Lastly, within the user manual it explains how the system is used for proper use and error prevention.

Introduction

The purpose of *Project 2* is to gain experience through developing, implementing, analyzing, and/or applying the things discussed within Software Development I. Based off of that, *Type It* is a text editor project. It is a fully java programmed text editor, similar to Microsoft's *Notepad* application, though with a few other aspects which makes it unique. This document explains the current operations of the system, physical requirements of the system resources, other applications currently available, and a brief user manual for using the system.

Detailed System Description

The application, *Type It*, is no different than a text editor. It has a text area in which a user can input text and save that text as a .txt file. This program is fully programmed using java.swing and java.awt. Along the toolbar of the text editor, there contains four tabs, each of

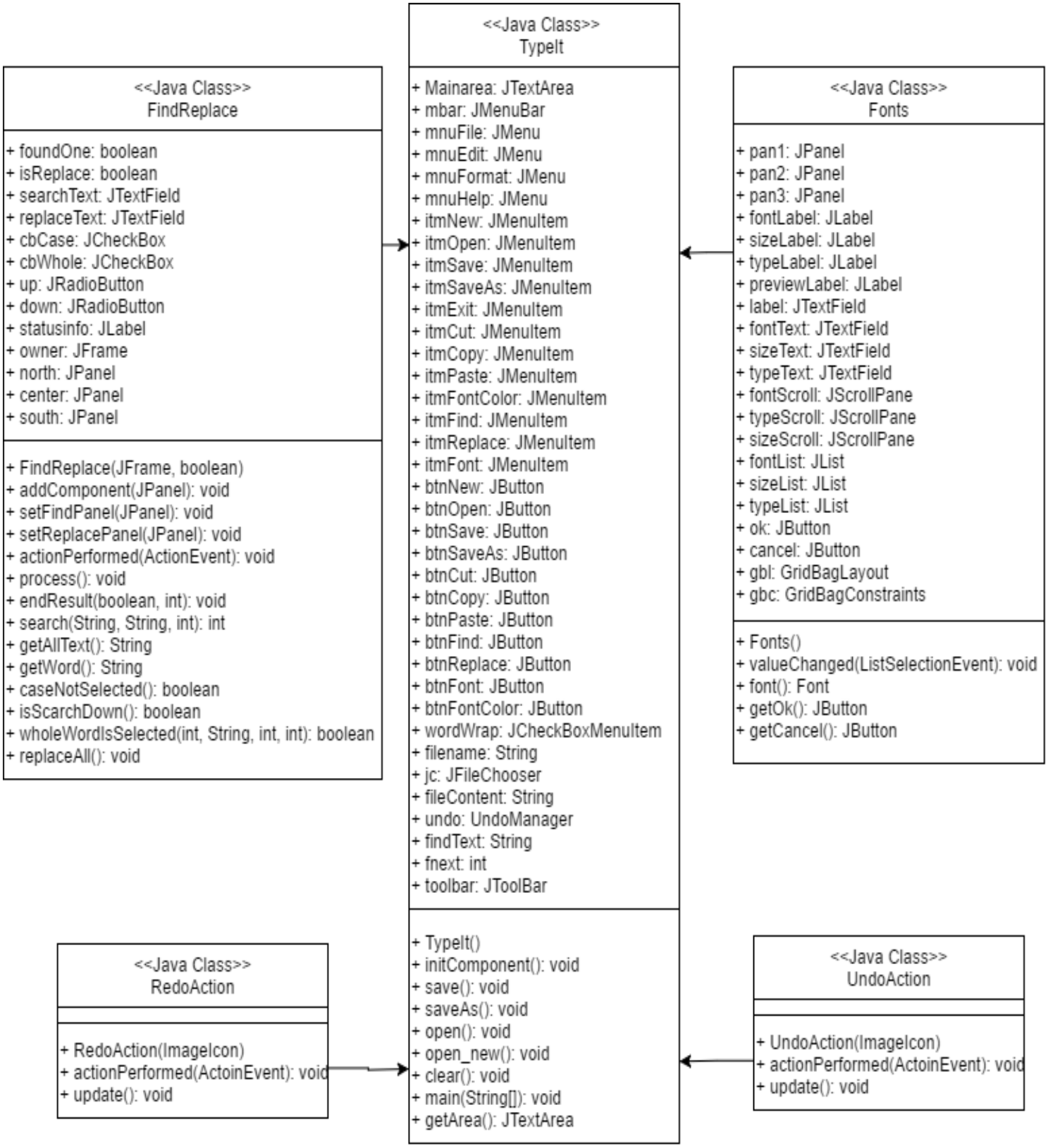
which contain sub functions listed as menu items. The toolbar tabs are *File, Edit, Format,* and *Help.*

The first tab, *File*, contains *New, Open, Save, Save As,* and *Exit.* The *New* menu item allows for a new file to be opened. When a file is currently open though not saved, an alert will appear prompting the user as to whether or not they would like to save the current file. The *Open* menu item opens previously saved text documents, the *Save* item saves the documents and the *Save As* allows for the user to save the current text file under a specific name and/or specific location they want to save the file.

The second tab, *Edit*, contains the *Undo, Redo, Cut, Copy, Paste, Find,* and *Replace*. These menu items are mainly self-explanatory. The *Undo* and *Redo* allows for the user to undo and redo any text, the *Cut, Copy,* and *Paste* is for copying text, pasting any copied text, and cutting the text so it could possible by pasted elsewhere. The *Find* and *Replace* functions contains a class of their own. The FindReplace.java class allows for a user to find a specific character or word within the document and to replace that with another character or word. This is for ease of access so that a character, word, or piece of text can be quickly found within a large text file.

A detailed visual of the java classes utilized for this application is below in the UML diagram.

***Type It* UML Diagram**



Requirements

The only requirements within system resources is for the system to have a current version of java and javac so that the program can run. It is currently being compiled and run through the command line. Extremely little system data or memory is necessary for this application, though when a text file is saved, it does take up memory by a minute amount.

Literature survey

Other available applications similar to *Type It* include, but is not limited to Microsoft Notepad and Notepad++. Microsoft's Notepad is the most similar to the application while Notepad++ is far more advanced as it is also the application being used for the programming of *Type It*. What make *Type It* different from other applications is that this application is has less functions than the other programs and may contain possible malfunctions that it was not tested for. Overall, *Type It* is a model of Microsoft's Notepad though with icons beside each function and the additional ability to change the color of the text.

User Manual

To run the application as it is, the user needs to know how to use the command line, or to have and IDE which can compile and run the program. Once the program is running, it opens up a white text editor with a menu bar. On the menu bar, the functions available under the *File, Edit,* and *Format* tabs are available on the toolbar in the form of their icon image. The toolbar items can be used either by their corresponding shortcut key or by pressing the icon buttons on either the toolbar or under the tabs on the menu. For example, to save a document, it is possible to use Ctrl+s, press *Save* under *File* on the menu, or press the save icon on the toolbar. Either option works in saving a text document. Overall, using the text editor is mostly self-explanatory.

The user can type text into the white space called the text area. Any typed text can be saved, while new documents can be opened, and previously saved documents can also be opened. The font color and font type for the over text can be changed by using the *Font* and the *Font Color* functions found in the menu and on the toolbar. When changing the font and font color, it is changed for the entire document. The function does not allow for a selected element to be changed only. The find and replace functions are for finding elements within the text documents and replacing those elements if needed. Also, the word wrap option makes it so that the text is formatted to fit the window of the text editor. A user can either check or uncheck the box for word wrap to format the text. The *Help* option on the menu does not have any function and is mainly there for show.

It is unexplainable, but for some unknown reason, two text editor windows open when the program is run. The best way to use the application in such a state is to minimize the first window which opened and use the one left unminimized. Closing one window closes the entire application. Other than this issue, the application performs well for simple uses. Opening large documents with this text editor may cause some errors when changing the font. There may be other errors present, but to prevent the ones which are known, it is best to follow these guidelines.

Conclusion

The programming of the text editor, *Type It*, was an experience in fully utilizing the java language of programming in further improving skills in programming. *Type It* utilizes three classes, TypeIt.java, FindReplace.java, and Fonts.java. These three classes are fully programmed in java using java.swing, which is essentially a GUI widget toolkit for java. The detailed system description particulars the ins and outs of the classes along with the UML diagram. The

application requires little system requirements and within the Literature Survey, it is pointed out that this application is similar to others currently available, specifically in relation to Microsoft Notepad. The overall how-to for the application is entirely described within the User Manual, including a few error prevention guidelines. A text editor is a simple application that nearly all computer users utilize whenever using a computer. Text editors are such a familiar application for most users that it usually goes unnoticed how useful the application is without any regard to the programming which goes into it. This project of creating a text editor wholly improves programming skills, specifically in java, making it a wholesome project to complete.

Works Cited