

Transcended C Editor

Rinu Rani Jose^{1*}, Niya Charles², Bessy Anna Varghese³ and Alakananda⁴

^{1,2,3,4}St.Thomas College of Engineering and Technology, Chengannur

Available online at: www.ijcsonline.org

Received: 22/Jun/2016

Revised: 10/Jul/2016

Accepted: 16/Aug/2016

Published: 31/Aug/2016

Abstract— Transcended C editor is a software that edits plain texts and comprises of Java Swings and AWT. This project has all the frames prepared in Swing. The Transcended C editor enables us to code C programs more efficiently. This editor is provided with features such as auto completions, signature help, syntax highlighting, string searching, descriptions, different font styles, sizes etc, along with an added option for compilation and run which makes it different from other editors.

Keywords— IntelliSense, AWT, GUI, GCC.

I. INTRODUCTION

A text editor is used to edit plain text files. Text editors differ from word processors, such as Microsoft Word or WordPerfect, in that they do not add additional formatting information to documents. We might write a paper in Word, because it contains tools to change fonts, margins, and layout, but Word by default puts that formatting and layout information directly into the file, which will confuse the compiler. If you open a .doc file in a text editor, you will notice that most of the file is formatting codes. Text editors, however, do not add formatting codes, which makes it easier to compile your code. Text editors have a feature set different from that of a traditional word processing program. The features of text editors vary from implementation to implementation, but there are several kinds of features that most editors have. Some of the useful features of this project are Syntax highlighting, IntelliSense which includes Completions and Signature help, String searching, descriptions on keywords, different fonts and font size.

After the source code written in this editor, we can compile and run the program using the GCC compiler (GNU compiler collection). If there are no errors, the command prompt window will be open and there we can type the statements for compilation and execution. In the case of errors, a new tab will be open showing the errors.

Thus the main objective of this project is to develop a text editor which is an enhancement to the existing C editors. This Transcended C editor enables us to code C programs more efficiently. This editor has features such as auto completions, signature help, different font styles, sizes, syntax highlighting, string searching, descriptions etc. Along with an added option for compiling and run which makes it different from other editors.

II. PROPOSED SYSTEM

TurboC is the most commonly used editor and compiler for C. But this does not provide features like prediction, completion, signature help, fonts, etc. Transcended C Editor provides these features which take it to another level. Thus enables to code C programs more efficiently. This editor is developed in Java. The main features proposed in this editor are Syntax Highlighting, IntelliSense, String searching, Descriptions, different fonts and font size.

III. FEATURES

A. IntelliSense

IntelliSense is the general term for a number of features: List Members, Parameter Info, Quick Info, and Complete Word. These features help you to learn more about the code you are using, keep track of the parameters you are typing, and add calls to properties and methods with only a few keystrokes. IntelliSense exposes results of analyzing your programs in two different forms: completions and signature help.

(a)Completion

Auto complete, or word completion, is a feature in which an application predicts the rest of a word a user is typing. In graphical user interfaces, users can typically press the tab key to accept a suggestion or the down arrow key to accept one of several. Auto complete speeds up human-computer interactions when it correctly predicts the word a user intends to enter after only a few characters have been typed into a text input field.

Auto complete of source code is also known as code completion. In a source code editor auto complete is greatly simplified by the regular structure of the programming languages. It involves showing a pop-up list of possible completions for the current input prefix to allow the user to

choose the right one. Therefore, auto complete then serves as a form of convenient documentation as well as an input method.



Figure 1. Prediction

When user begins a line which starts with a frequently used phrase, the editor automatically completes it, up to the position where similar lines differ, or proposes a list of common continuations (prediction).

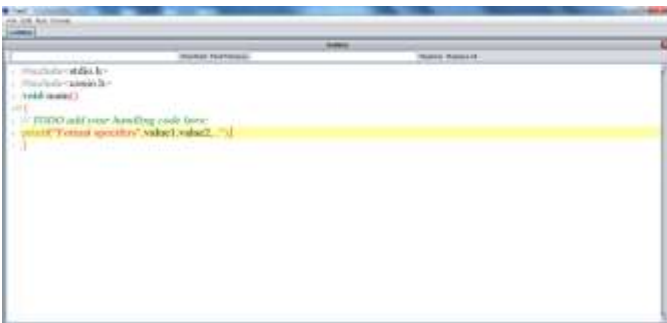


Figure 2. Completion of the selected keyword

(b) Signature help

Signature help is displayed when writing a function call, and includes any documentation and parameter information that is available. Signature help appears automatically after typing the '(' in a function call, and can be displayed at any time by pressing Ctrl+Space inside a function call.

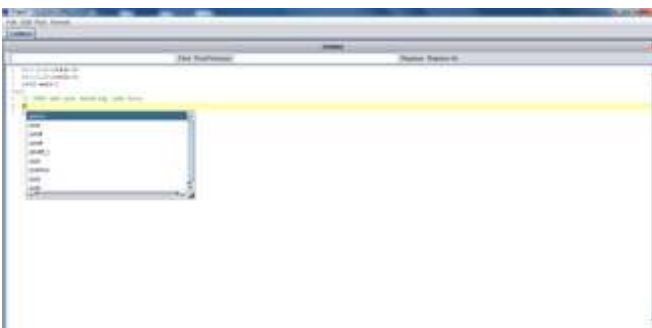


Figure 3. Signature help

B. Syntax Highlighting

Syntax highlighting makes it much easier to read and write in structured languages since syntax errors and other elements become visually distinct with color-coding. This feature is solely for the aid of human editors, since it does not affect the meaning if the text in any way. Highlighting improves readability without altering the text itself.

Some of the staple syntax highlighting functions includes brace matching, which locates and highlights matching or missing opening and closing tags, and highlighting string literals, which make spotting delimiter errors much easier.

Syntax Highlighter is a feature of this Transcended C editor that is used for programming. The feature displays text, especially source code, in different colors and fonts according to the category of terms. Highlighting does not affect the meaning of the text itself; it is intended only for human readers. Syntax highlighting is one strategy to improve the readability and context of the text; especially for code that spans several pages. The reader can easily ignore large sections of comments or code, depending on what they are looking for. Syntax highlighting also helps programmers to find errors in their program. Syntax Highlighter is a fully functional self-contained **code syntax** highlighter developed in JavaScript. Contextually highlights source code and other text that appears in an organized or predictable format.



Figure 4. Syntax highlighting

C. Descriptions

Providing description about the in build functions and methods, which will help the user to understand and follow the syntax and arguments lists effectively.



Figure 5. Providing description about a keyword

D. Fonts and font size

This editor provides all fonts that can be used to make a text more appealing. Along with the different types of font, the style and size of each selected font can also be changed.

This feature seems to be very effective while taking classes using projectors to teach C programming.

To select custom fonts, choose **Format**, then click on **Fonts**, and click the font name to see a list of available fonts, style to see available styles and size to see various sizes.



Figure 6. Dialog box showing different fonts

E. Find and Replace

Finding that one, specific keyword in an entire file can be difficult, especially if needed to locate all instances of the word and change it to something else. Fortunately, many text editors come with advanced Find and Replace functions that allow you to search for whatever you need and then replace it with another phrase. So this feature is included in this editor which makes it more efficient.



Figure 7. Finding a string

The string needed to be searched can be written in the text area provided with the find menu and then click the find option, which highlights all occurrences of the searched string. After finding a string it is also possible to replace it with another string.

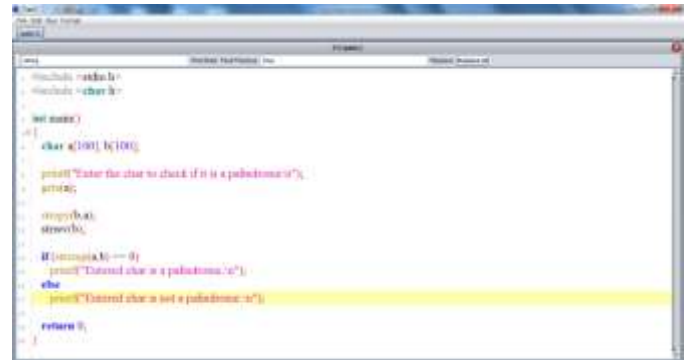


Figure 8. Replacing a string with another string

F. Compilation and Run

Also this editor has a compiler linked to it which allows the compilation and execution of the code to be done here itself. In this editor we are using GCC compiler to run the C programs coded in the text editor. A function call in the source code links to this compiler. The GNU Compiler Collection includes front ends for C, C++, Objective-C, FORTRAN, Java, Ada, and Go, as well as libraries for these languages (libstdc++, libgccj). GCC was originally written as the compiler for the GNU operating system. The GNU Compiler Collection (GCC) is a compiler system produced by the GNU Project supporting various programming languages. GCC is a key component of the GNU tool chain. The Free Software Foundation (FSF) distributes GCC under the GNU General Public License (GNU GPL). GCC has played an important role in the growth of free software, as both a tool and an example.

Here after the source code is written in the editor, we can compile and run the program using this GCC compiler by selecting the **compile and run** option provided in the **run** menu. If there are no errors, the command prompt window will be open and there we can type the statements for compilation and execution. In the case of errors, a new tab will be open showing the errors.

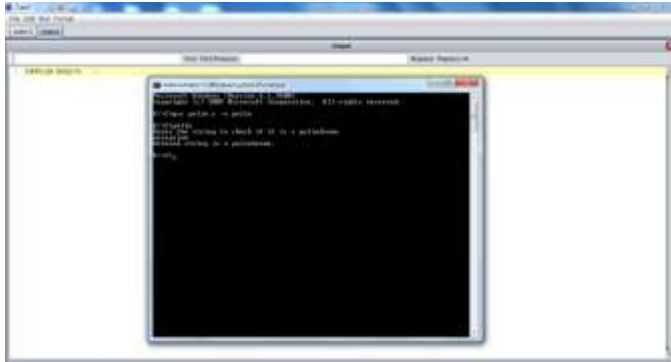


Figure 9. Execution of the program

G. Other Miscellaneous Features

Other miscellaneous features include different file menu options and edit menu options.

(a) File Menu options

The file menu contains options to open a new file or an already existing file, to save an open file in the current workspace, to create a copy of an open file under a new file name and to close the editor.

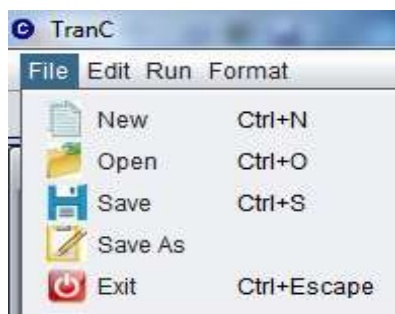


Figure 10. File Menu

(b) Edit Menu options

The edit menu contains undo and redo options to repeat the last command called, options to remove the selection and move it to the clipboard, options to create a copy of the selection, and option to add a selection to the active document, beginning where the cursor is located, this command is available only if you have already used the **Cut** or **Copy** commands.

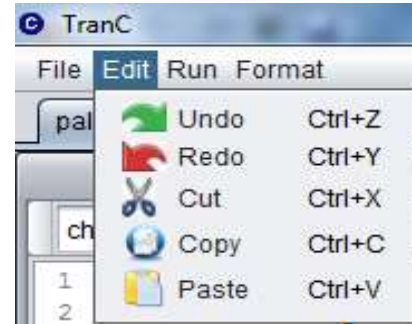


Figure 11. Edit Menu

IV. CONCLUSION

Nowadays everyone is likely to work with Java even though the syntax of this programming language is not simple as compared to others. This is only because of its added features. So by this editor we are enhancing the capabilities of existing C editor in order to make it a language competent to Java. This TranC or Transcended C Editor has many features which makes it easy to write codes. Also this editor has a compiler linked to it, that allows the compilation and execution of the code to be done here itself which makes it different from other editors.

REFERENCES

- [1] Cay S Horstmann and Gary Cornell, "Core Java-Advanced Feature," Prentice Hall, Vol.2, 9th Edition.
- [2] Herbert Schildt, "The Complete Reference Java," Tata McGraw-Hill Education, 7th Edition.
- [3] E Balagurusamy, "Programming in ANSI C" Tata McGraw-Hill Education, 6th Edition, Jan-2012.
- [4] Manuel D. Rossetti, "Java Simulation Library (JSL): An open-source object-oriented library for discrete-event simulation in Java", Int. J. of Simulation and Process Modelling, Vol. 4, 2008, No.1 pp. 69 – 87.
- [5] Yu Yan, Nakano Hiroto, Hara Kohei, Suga Shota, AiguoHe, "A C Programming Learning Support System and Its Subjective Assessment", IEEE International Conference on Computer and Information Technology, ISBN: 978-1-4799-6239-6, Sept 11-13, 2014.
- [6] Vamsi Krishna Myalapalli, Sunitha Geloth, "High performance JAVA programming", International Conference on Pervasive Computing, ISBN: 978-1-4799-6272-3, April 16, 2015.