User Manual
Parametrized String Matching Implementation for Software Plagiarism Check*

Supervised By : **Prof. Srinivasaraghavan G**

Team
amit.tomar
(MT2013008)

siddhesh.dosi
(MT2013150)

srinivas.r.vaidya
(MT2013152)

@iiitb.org

26 - July - 2014

*Although the official name of this project is *Parametrized String Matching Implementation for Software Plagiarism Check*, internally within the team we had code named it as the ***CopyDog***. Any reference to CopyDog in this or other documents shall be understood to as referring the same project only.

1 Launching the application

If all the dependencies (mentioned later in the document) are met properly, application can be launched as follows :

1. Change the directory to the folder containing your application.
2. Execute the command

   ./CopyDog

   Change the permissions using *chmod* command, in case you get permissions related errors.
3. You can also execute it by double clicking using a GUI.

2 Using the Application

CopyDog works in two mode of operations. Using one option you can select all the files in the selected folder, with the selected language extension, automatically. In the second option, you can select the files manually to check them for plagiarism. In the *Startup Screen*, you can select the language you want to check plagiarism on, and the minimum number of characters that should be considered for any plagiarism related case. Then you can click on of the following options :

Select all files in the folder

Clicking on this option, opens up a folder browser which allows to select all the files in that folder, of the extension corresponding to the language specified by you, and generates the plagiarism related information.

Select files manually

Clicking on this option, opens up a file browser which allows to select files manually in that folder, and generates the plagiarism related information.

Export and Exit

CopyDog also provides an efficient way to save the plagiarism related information in persistence. You can simply click on the save and exit button. CopyDog will save the information in your current working directory with name:
CopyDogPlagiarismInformation_[TimeStamp].

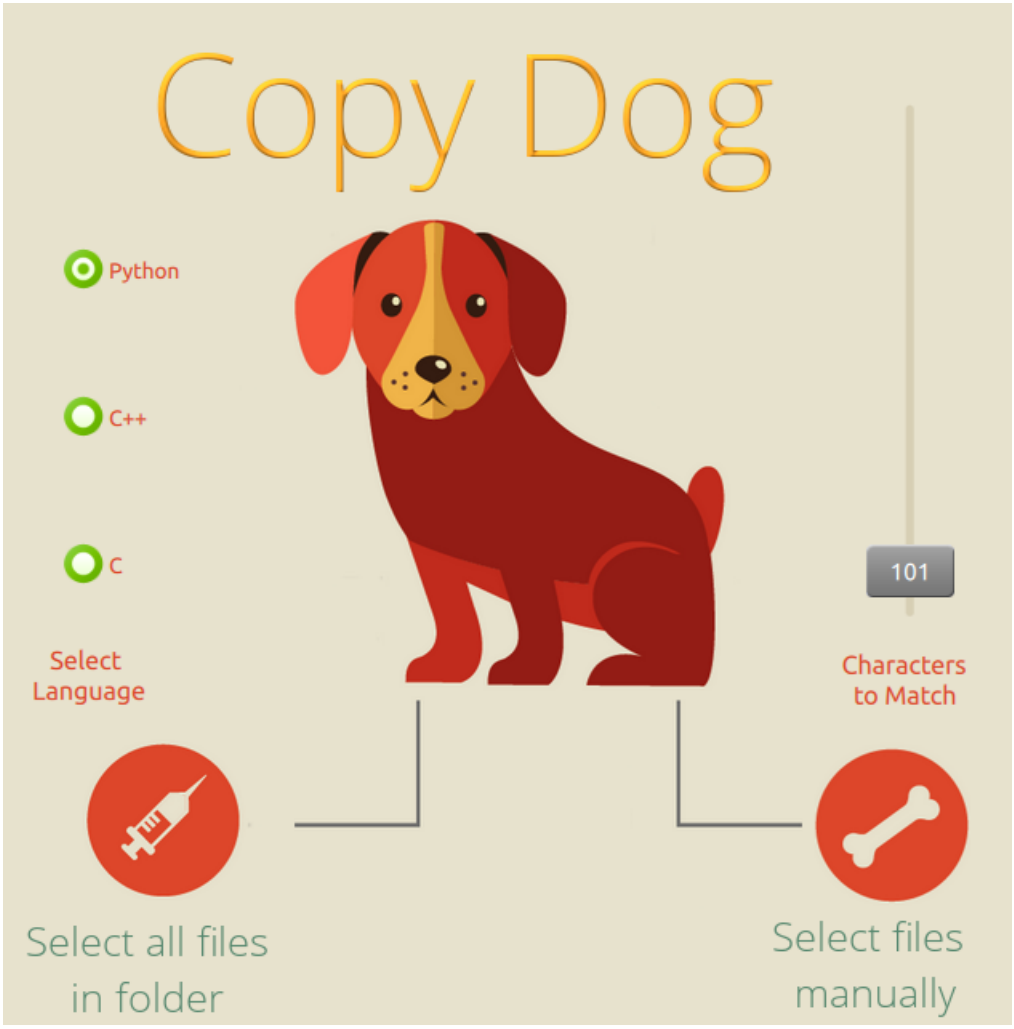

Figure 1: Main Screen of CopyDog

Figure 2: Screen showing Plagiarim information

3 Dependencies and Build

CopyDog has been built using Qt-5.3. Though there should not be problems with any version of Qt greater than version 5.0. The uploaded built version is for 32-bit Intel platform. Any other platform will require a rebuild of the project. *Src* folder in the code contains a *.pro* of Qt, which can the opened up using the Qt-Creator IDE for editing/compilation purposes. In case the IDE is not available, the Makefile present in the *Build* folder can also be used for a re-build. Try doing a clean-build to avoid any mismatch in the library version of the obj files already present in the Build folder. Python version 2.7 has been used for the Python code.

Following files whould be present in the same folder as the executable :

1. CopyDog.png
2. MainScreen.png
3. slider.png
4. radioChecked.png
5. radioUnchecked.png
6. main.qml
7. PythonParser.py
8. Unparse.py

4 References

Following sources were referred before writing this documents to get the basic understanding of Kernel development process and approximate time it might require:

1. **Images**

   Dog image present on the home screen has been taken from the work of Marina Zlochin. (*Usage request for this image is still pending with her.*)

https://www.behance.net/ma_rish

2. **Python Code**

   For converting the Abstarct Synax Tree structures of Python code to the normal source code of Python, we have used an *Unparser* from this site :

   http://svn.python.org/projects/python/trunk/Demo/parser/unparse.py