**GUCC development team**

**Final Report**

**Objective**

The objective of this directed research project was to identify and fix the problems related to the usability of the graphical user interface (GUI) of the code counter, and to provide support for all the languages supported by the command line version.

**Project Requirements**

1. Microsoft Visual Studio 2010.

2. QT-Version 5.0.x

3. QT Add-in for Visual Studio

4. QT-Creator (Linux Platform)

**Problems Identified**

The various issues found within the original GUI were:

1. The extensions were hard-coded. Dynamically adding extensions was not possible

2. Menu items were just dummy buttons with no action.

3. Multiple-selection of files to count was not possible.

4. Multiple-removal of files was not possible.

5. No information displayed regarding the events happening in the background.

6. UCC result window cannot be minimize, OK/Cancel button does not work

7. Wrong files shown on output when ASCII is triggered.

8. Several buttons had no purpose

9. Missing ‘About’ file in the repository.

10. The tool did not work on Linux systems.

**Fixed Defect:**

1. The extensions are being read from a file. It is now totally dynamic and can provide support to any new languages added to the command line version.

2. All the menu items have been checked and re-arranged to show valid contents.

3. Multiple-selection of files (also included for differencing) and multiple-removal of the selected files has been made possible now.

4. All the processing information is now being displayed in a status bar at the bottom of the application. The process which used to take a long time to respond, now gives the information about the things happening in the background.

5. Pending issue, still require progress bar.

6. UCC result window can be minimize, OK button works and Cancel button is deleted

7. The wrong output files which were being shown previously with the -ASCII option have now been removed. Only the right ones are now shown in the proper window.

8. The buttons which were not needed for the application have been removed. The buttons which are required for the application have been suitably added.

9. The user manual has been updated and made available in the repository.

10. The tool has been made cross-platform. It now runs successfully on a Linux-OS as well.

**Challenges faced**

1. In making the tool cross platform, we needed to take care of specific OS properties (using define macros).

2. Creating the executable for Windows and Linux versions.

3. Missing files need to be installed (might need to install -mesa packages for Linux etc).

   Possible issues: “glu” missing when compiling under Linux

   Solution: this is related to the OpenGL library which the possible solution is to install mesa and mesa-dev packages

**Upgrades to the existing project**

1. All the languages which are present in the command line version of the code counter are now present in the GUI version as well. In future, if a new language is to be added to the command line version, it will also be supported in the GUI version.

2. The GUI is now cross platform. It can be run on Linux and Windows operating systems. The code changes for the same have been made at required places.

**Future Development Options**

1. Develop a visual differencing tool for the GUI version of the project.

2. Develop an install packet which could install the software on any system, without having to compile it on every machine.

3. Reorganize and rewrite the user menu:

Suggestion: the framework of GUCC User Manu

Part1: Introduction of UCC project and GUCC

Part2: How to setup GUCC for windows/Mac/Linux platform

Part3: Usage and functionality of GUCC

Part4: How to develop GUCC project: how to download source code, set up IDE, compiling source code and enter debug mode (Please check Appendix 1. Compiling Guide)

**Team Member**

2013 Spring Semester: Xunnan Xu, [Pulkit Maheshwari](javascript:parent.addSender(%22%20Pulkit%20Maheshwari%20%3cpmaheshw@usc.edu%3e%22)), [Sanath Manavarte](javascript:parent.addSender(%22Sanath%20Manavarte%20%3cmanavart@usc.edu%3e%22)), Yuzhe Wu

**Appendix**

1. Compiling Guide

**Windows Platform (Visual Studio + QT)**

Step1: Go to website <http://qt-project.org/downloads> to download the QT application framework.

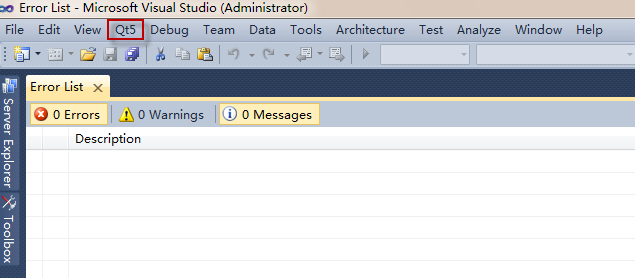
[Qt 5.0.2 for Windows 32-bit (VS 2010, 485 MB)](http://download.qt-project.org/official_releases/qt/5.0/5.0.2/qt-windows-opensource-5.0.2-msvc2010_32-x86-offline.exe)

Step2: Install Microsoft Visual Studio 2010

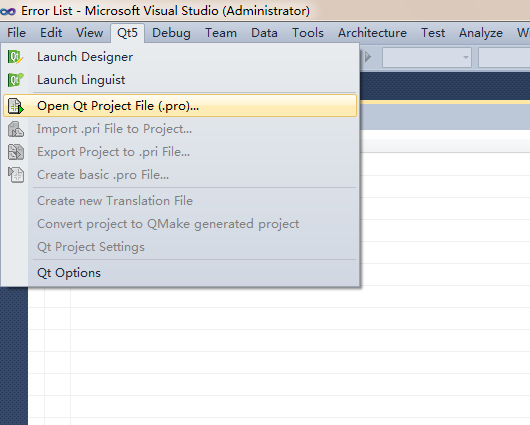
Step3: Download QT version studio add-in. Go to website <http://qt-project.org/downloads>, download [Visual Studio Add-in 1.2.0 for Qt5](http://releases.qt-project.org/vsaddin/qt-vs-addin-1.2.0-opensource.exe).

Step4: Download the source code of GUCC project

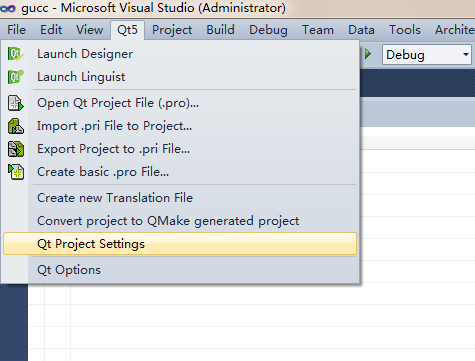
Step5: Open Microsoft Visual Studio, if the add-in was successfully installed, the QT drop-down menu will appear at the middle of menu bar of Visual Studio.



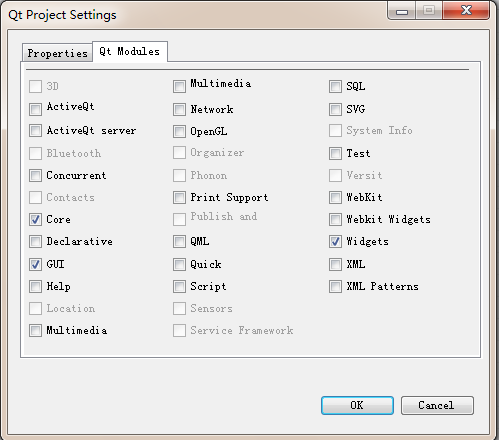
Step6: Click Qt5->Open Qt Project File (.pro)…, select GUCC to open



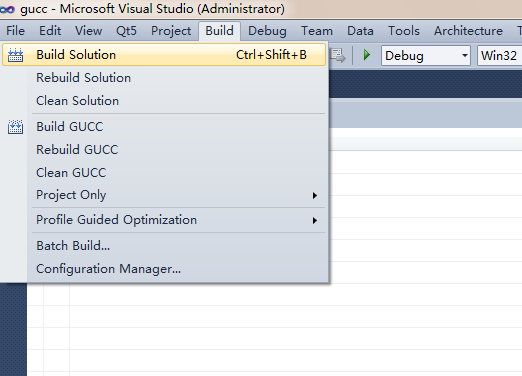
Step7: Open Qt5-> Qt Project Settings



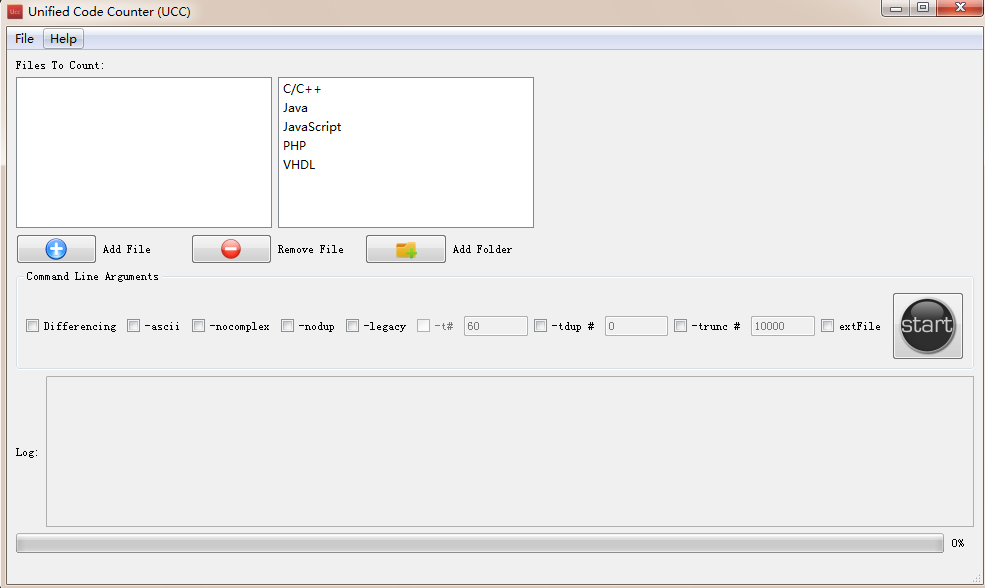
Step8: In Qt Modules, select Core, Widgets and GUI.



Step9: Select Build->Build Solution



The debug mode of GUCC is successfully activated.



**Mac OS Platform (QT Creator)**

Step1: Download and install Qt for Mac at <http://qt-project.org/downloads>

Step2: Download Xcode Command Line Version

Step3: Load the project into Qt creator.

D:\Documents\Qt\Snapshot\1.tiff

Step 4: Switch to “Release” mode in the left pane. Then click Build -> Build All. Qt Creator by default will use clang as the complier. If somehow this fails, see Step 5

D:\Documents\Qt\Snapshot\2.tiff

Step 5: Click Projects->Manage Kits->Kits and add GCC as the compiler

D:\Documents\Qt\Snapshot\3.tiff

Step 6: Compile again as in Step 4.

Step 7: Click “Run” to see if the GUI works.

Step 8: Use macdeployqt to build the app bundle for Mac: macdeployqt is located in the bin folder under Qt directory.

Here assume it is at ~/Qt5.0.2/5.0.2/clang\_64/bin/macdeployqt

Switch to the GUI project folder and execute

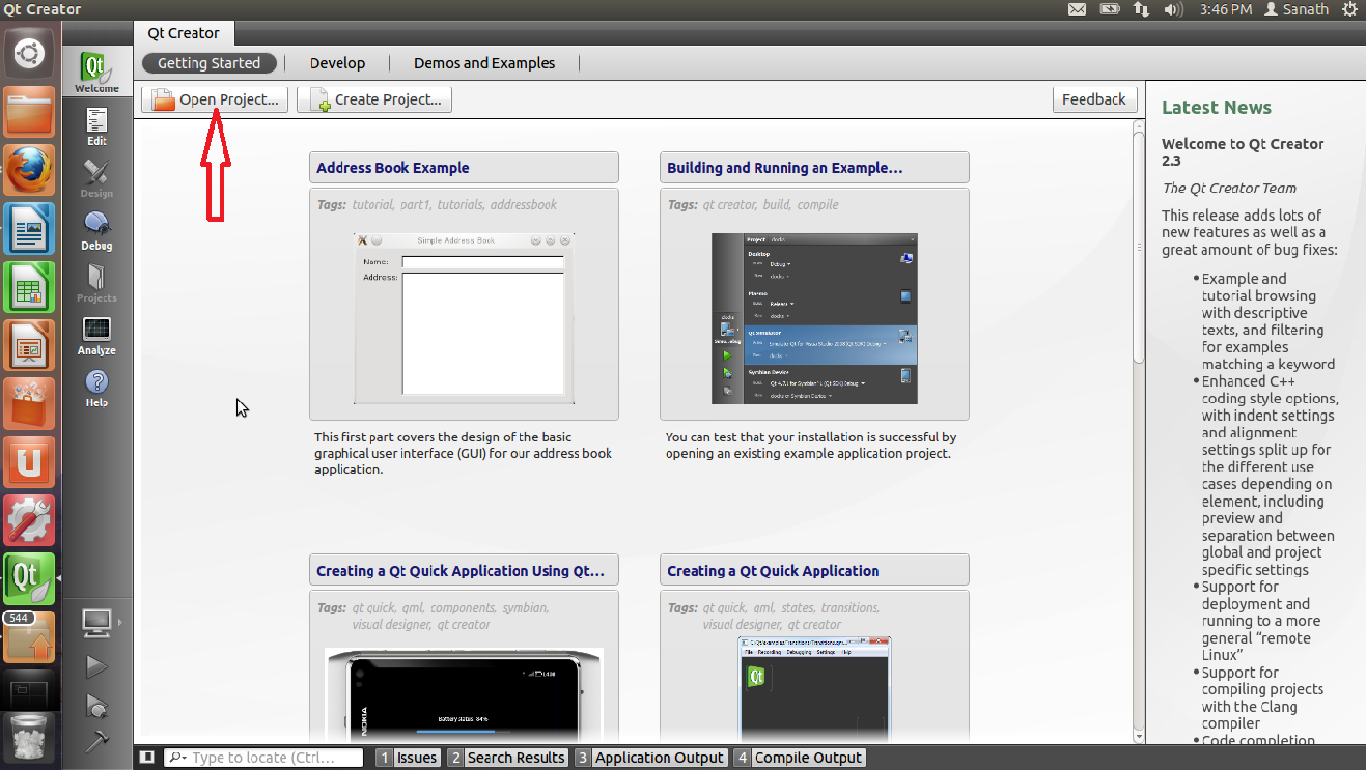
sudo ~/Qt5.0.2/5.0.2/clang\_64/bin/macdeployqt GUCC.app -dmg -verbose=2

Note: using sudo to elevate user privilege is not necessary but recommended.

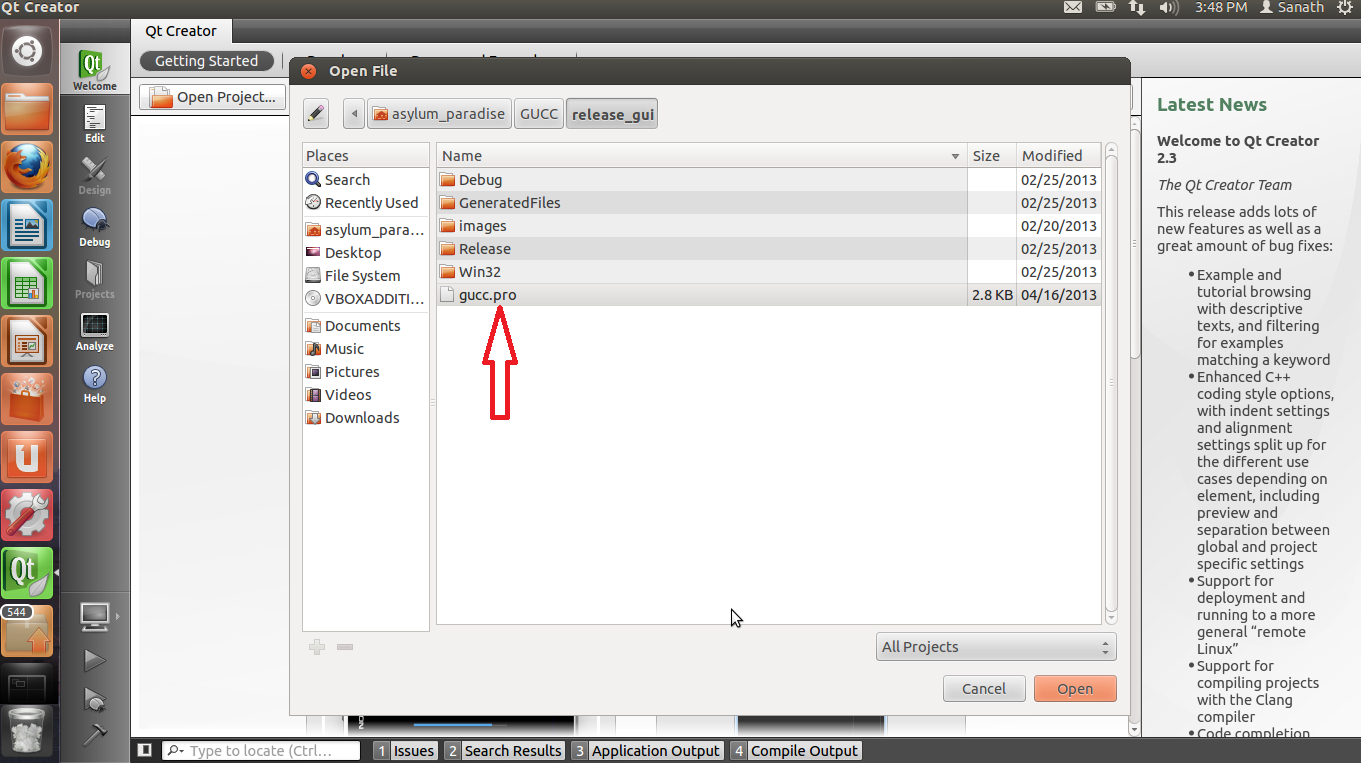
**Linux based platforms (QT Creator)**

Step 1: Go to <http://qt-project.org/downloads> and download and install QT creator, whichever one is applicable to the version of the operating system being used.

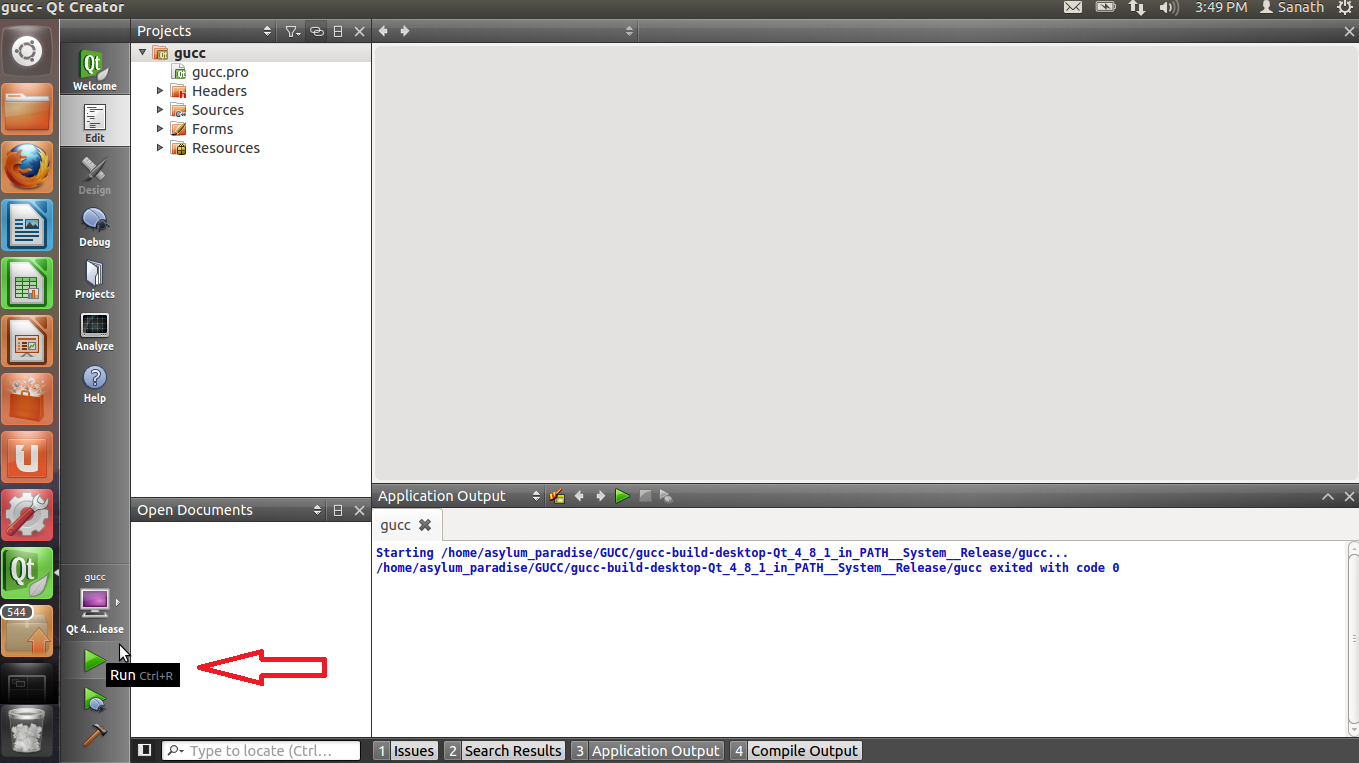
Step 2: Start QT creator. The application would be as shown below. Click on ‘Open Project’ button.



Step 3: Place the folders containing the GUI code and the command line version code within the same folder. Locate the file ‘gucc.pro’ from the GUI code as shown below



Step 4: Click on the ‘Run’ option to compile and run the program.



Step 5: You should be able to see the code counter software up and running after it compiles.

