

CS3249 User Interface Development
AY2014-15 Semester 2

Assignment 1: MyEditor
Deadline: Wednesday 18 Feb 2015 12pm

This assignment continues from Lab 1. The objective is to maintain a list of the names of recently opened files in the File menu. This way, the user can more easily open these files.

Program Requirements

1. The editor should be complete with the features worked out in Lab 1:
 - File → New, Open, Save, SaveAs.
 - Edit → Cut, Copy, Paste.
 - Display current file name in window title. Show * symbol in window title if file is modified.
2. Set the recent file list to a maximum of 5 file names.
3. The recent file list should be initially empty when the editor is launched. There is no need to save the recent file list.
4. When a file is opened, its file name is inserted at the top of the recent file list and labelled 1. Previously opened files are listed in decreasing order of file opening sequence and labelled 2 onward in creasing order. For example:
 1. lab exercise.txt
 2. my document.txt
 3. other stuff.txt
5. Make sure there is no duplicate in the recent file list.
6. Selecting a file from the recent file list to open moves its file name to the top of the recent file list.
7. Keep a record of the current directory of the current file, which is initialised to “.”. If the user opens a file from a different directory and the file is successfully opened, then the new directory is set as the current directory.

Assignment Tasks

You may use the following ideas to implement the list of recently opened files in the File menu:

1. recentFiles

- Add a variable called `recentFiles` of type `QStringList` to `MyEditor` class for keeping the file names of recently opened files.
 - `QStringList` is a `QList` of `QString`. It is a convenient class for maintaining a list of recently opened files.
2. `recentFileActions`
 - Create an array of `QAction` objects called `recentFileActions` for storing the action associated with each recently opened file.
 - Set each action's visibility to `false` to hide them initially.
 - Connect each action's triggered signal to a new slot function called `openRecentFile()`.
 - Add these actions into the File menu. It is a good practice to add a separator between them and the other menu items.
 3. `openRecentFile()`
 - This is a new slot function.
 - Use the following statement to identify the action that is triggered:

```
QAction *action = qobject_cast<QAction *>(sender());
```

`sender()` is a function inherited from `QObject` that returns the `QObject` that sends the signal.
 - If `action` is not `nil`, open the file whose full file name is stored in `action`.
 4. `updateRecentFiles()`
 - Update the file names in the `recentFiles` list.
 - Update the actions in `recentFileActions` with the file names in `recentFiles` in the same order.
 - Use `setText` to set the action's text to file name, including file suffix but excluding its path name.
 - Use `setData` to set the action's data to full path name. This data will be used later when this action is selected by the user to open a recent file.
 - Set the action's visibility to `true`.

If `recentFiles` is shorter than `recentFileActions`, then hide the remaining actions in `recentFileActions`.
 5. `open()`
 - Revise it to get a file name from the current directory.
 6. `setCurrentFile()`
 - Revise it to also set the current directory if a file is successfully opened.

Note

For this first assignment, you may refer to Chapter 3 of *C++ GUI Programming with Qt 4* for an example of how to implement recent file list in File menu. Understand the technique, then implement in your program.

Submission

1. Demonstrate your program to the TA, who will instruct you to go through several test cases.
2. If your program has errors, you may correct it and present to the TA for a second time before deadline.
3. If your program is correct in the first or second presentation, you get full marks. If you program is still incorrect at the second presentation, you get half marks. If you don't submit the assignment, you get no mark.