Virginia Tech
Bradley Department of Electrical and Computer Engineering
ECE-3574: Applied Software Engineering Spring 2016

Homework 4

Submission Details

You must submit the solution for this homework in electronic form through Scholar (under ECE3574 → Assignments → Homework 4). The submission must be a gzipped tar file (.tar.gz) with your source code. Include all necessary project files, but no binary or compiled files. Your program will be run to evaluate its correctness, and the source code will be reviewed for adherence to the Qt programming style. Your program must run on Ubuntu 15.10.0 and compile/build using the GNU C/C++ compiler and the qmake/make tools. The following information must be included at the top of each of your source files as comments: your full name, your student ID number, your email address, class (ECE 3574), and the title of the assignment (Homework 4). The submitted file must be given a name in the following form: ***LAST_FIRST_hw4.tar.gz*** where LAST is your last or family name and FIRST is your first or given name. Paper, email or Drop Box submissions will not be accepted. All work must be submitted by the announced due date/time. **Late submissions will not be accepted!** (Don't do it! You have been warned!)

Questions

Use the Homework 4 forum in the Discussion Board area of the class web site to ask questions about this assignment. Do not post questions that contain specific information about the solution.

Honor Code

As stated in the syllabus, in working on homework and projects, discussion and cooperative learning are allowed. However, copying or otherwise using another person's detailed solutions to assigned problems is an honor code violation. See syllabus for details.

Learning Objectives:

The primary learning objectives of this assignment include learning to use event-driven programming abstractions including the Qt event loop, use of graphical widgets and creation of graphical applications. Additional learning objectives include use of design patterns such as the composite pattern.

Homework:

The First Graphical Application

Write an application with four buttons (QPushButton) and a text edit box (QTextEdit) as shown in Fig 1.1

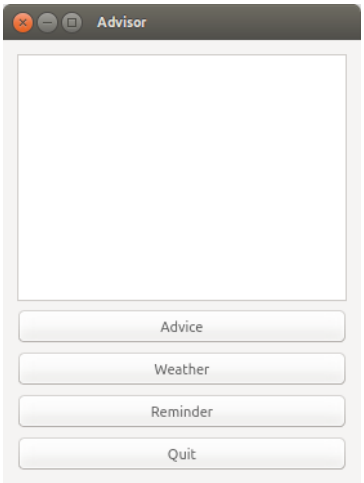

Fig 1.1

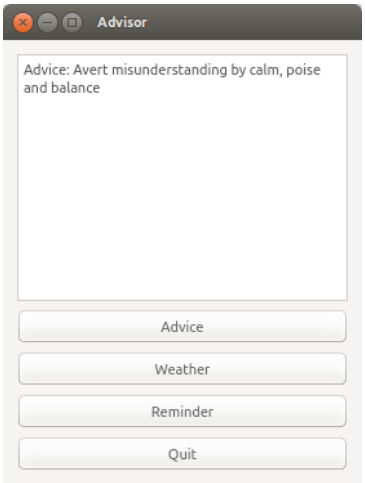

Fig 1.2

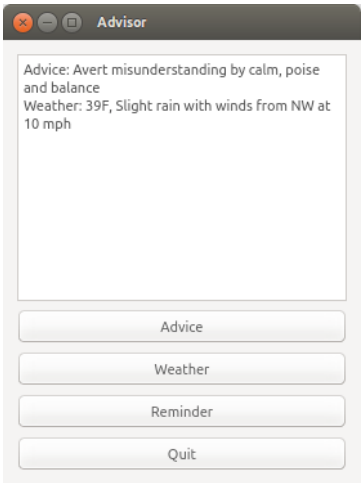

Fig 1.3

a) The first button should be labeled “Advice”. It should select a random piece of text from a file named “advice.dat” and append the contents to the QTextEdit window as shown in Fig 1.2. The word “Advice” should be prefixed at the beginning of the text (23 points).

b) The second button should be labeled “Weather”. It should select a random sentence about the weather from a file named “weather.dat” and append the contents to the QTextEdit window as shown in Fig 1.3. The word “Weather” should be prefixed at the beginning of the text (22 points).

c) The third button should be labeled “Reminder”. It should pop up a message dialog with a randomly generated (fictitious) meeting time and descriptive message from a file named “reminder.dat” as shown in Fig 1.4

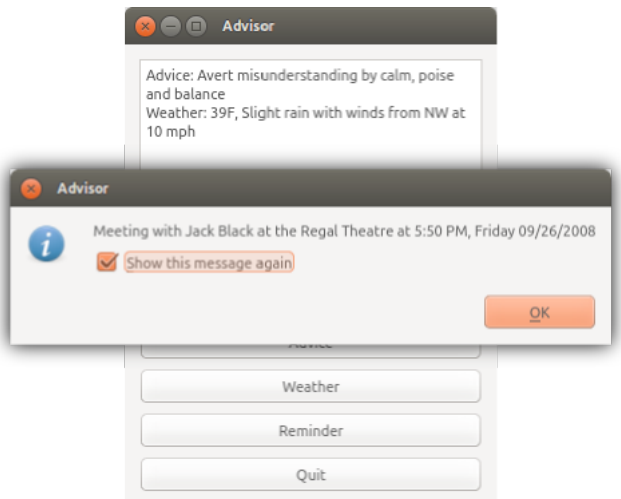

Fig 1.4

The pop-up window for the Reminder button should also have a checkbox which is marked when a message appears. When the checkbox is not marked it should not bring up the current reminder again until the next time the program is run. If there are no reminders that can be shown then pressing the reminder button should instead append a message indicating that there are no more messages to the QTextEdit window as shown in Fig 1.5. (35 points)

d) The fourth button should be labeled “Quit”. It should pop up a dialog box that asks the user to respond whether he or she really wants to quit the program. If the user selects yes, it should terminate the program. Otherwise it should close the dialog box and return to the program, as shown in Fig 1.6. (20 points)

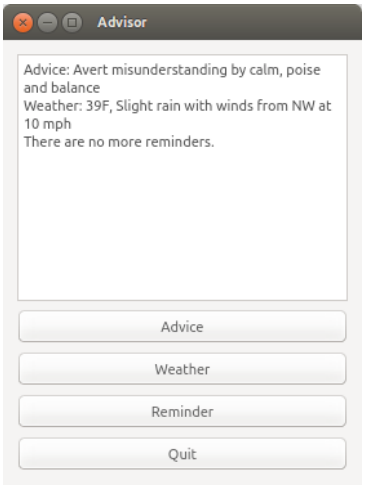

Fig 1.5

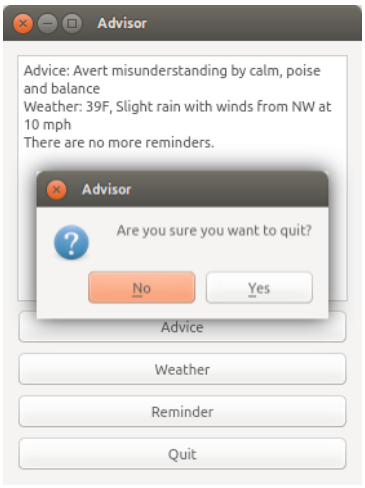

Fig 1.6

NOTE

1. You must use signals and slots to connect the button clicks with the appropriate functions

2. You must not use a graphical form editor like Qt Designer for this homework. You need to write all the widget interface code yourself.

3. The format of the files are: generally, one sentence/message per line. However, messages can span on multiple lines by having a \ (backslash) as the last non-whitespace character on each line in the file, except the last line. If the last line has such a backslash the error should be reported to stderr, the QTextEdit window or via a popup window. In other words, the trailing backslash character denotes the current message continues on the next line.

4. Pop-up windows must be modal. While a pop-up is displayed if the user presses a button in the main window it will not affect the state of the program.

5. **There is no partial credit among parts**. For example, if the Reminder button works as stated, you will receive 30. Otherwise it is a 0. Also, remember the credit covers your implementation. You will lose all the points even if your program works, but the code does not conform with the specification or the concepts taught in class.

6. In the next page you can find example *.dat files that your application should support. In the final evaluation these and other input files will be used to test your application, make sure that the application works even shuffling the different lines of the input files.

advice.dat

```
Adv\
ice \
0
Advice 1
Advice 2
Advice 3
Advice 4
```

reminder.dat

```
Rem\
inder \
0
Reminder 1
Reminder 2
Reminder 3
Reminder 4
```

weather.dat

```
Weather 0
Weather 1
Weather \
2
Weather 3
Weather 4
```