Assignment 1

Requirements

This first assignment should not be difficult. It does not have difficult GUI aspects to solve, but it does have a bit more challenging text processing. You are to write a program that reads its input data from a file named `.hwk1rc` in the current working directory[1]. The file contains the following information:

width a positive integer specifying the minimum width of the window in pixels, which might be larger if the title or the text forces it to be; it should not exceed 800.

height a positive integer specifying the minimum height of the window, which might be larger if the title or the text forces it to be; it should should not exceed 600.

title a string of at most 80 characters without newlines that will appear in the title bar of the window.

text a string that will appear in the content area of the window, centered on all sides by a margin of 5 pixels. It should not exceed 12 lines long.

and is in the format

```
width = positive-number
height = positive-number
title
text
```

You can assume there is at least one space character on either side of the "=" in the first two lines. For example, the file might contain the following data:

```
width = 80
height = 30
The First GTK+ Homework Assignment
This program was written by elves working into the
wee hours of the morning the night before it was due.
Special thanks to the gnomes who assisted the elves.
All rights reserved @2011.
```

The program is required to check all input data for validity, meaning that it must ensure that the numeric values are in range, and that the textual data is within the specified values. If any data is out of range, the program should terminate with an error message on the terminal window explaining why it quit unexpectedly.

Graphical Contents

The window should be a top-level window with a close box and should destroy itself and quit when the close-box is clicked. The text should be displayed within the content area of the window and should be selectable, so that the user can select any portion of it and copy it. Below the text there should be a button, aligned either to the left, or right or centered, as you see fit. The button should be labeled "`NEXT`".

When the user clicks the button, the text content should disappear and the button should be relabeled "`QUIT`". When the user clicks the button again, the application should quit.

[1] A real application would either read from a file in the user's home directory.

Style

This program, like all others that you write in this class, should be properly documented. Examples of proper documentation will be available in the course home directory. It should include a preamble containing the program name, author, date of creation, description, usage, and build instructions, and any other notes that you think the reader should see. Variables should be named clearly, following the GTK+ style of being verbose and descriptive, as should functions and any other symbols you define. You can write this in C or C++, but I suggest you do this in C, since otherwise you will have to read the gtkmm reference manuals online. Gtkmm is installed on all lab machines.

Hints

- There are various ways to parse the input file. It is designed to be easy to parse. You may use either the string processing functions in C's standard library (i.e., `scanf()`, `atoi()`, or `strtol()`, the string utility functions in GLib, or the regular expression functions of GLib. If you write in C++, you are free to use any of its `iostream` methods.
- You will need a label widget to display the text.
- You will need some kind of layout container.

Submission

Name this program `hwk1_`*`username`*`.c` or `hwk1_`*`username`*`.cc`, where *username* is the user name you have on our UNIX system. If you have a multi-file submission (as would be the case if you wrote a makefile), put all files in a directory named `hwk1_`*`username`* and zip it to the name `hwk1_`*`username`*`.zip` (or tar/compress to `hwk1_`*`username`*`.gz`). Put the submission into the directory

```
/data/biocs/b/student.accounts/cs493.70/projects/project1
```

on our file system. Make sure it is not readable by anyone but you (permission 700).