CISC 2000—Computer Science II
Fall, 2013

Programming Project # 0: Getting Started (the GUI version)

The purpose of this assignment is to ensure that you understand the workflow involved with using the author's GUI. Think of it as being a "Hello, world!" program that uses graphics.

Your task is to get the final example in Chapter 12 of the text to run; this is the example in Section 12.7.10. As we did last semester, each project in this course must have its own subdirectory; in fact, this is more crucial this semester than last semester. So the first step will be for you to make a directory `~/private/cs2/proj0` in which to do your work.[1]

Once you have done so, copy all the files (a `Makefile`, as well as two graphics files) in the directory

`~agw/class/cs2/share/proj0`

to your working directory. For now, you can treat makefiles[2] as being magical resources being handed to you by the Makefile Wizard (yours truly), although you're certainly invited to read about them (and I'll give you a general explanation of what they're all about).

You now need to create a file `proj0.cc` in your working directory, that contains the source in Section 12.7.10, as well as all the material that it back-references. If you want to type this in by hand, that's fine. Alternatively, recall that all the book's source code is someplace in the class "share directory" (which is `~agw/class/cs2/share`). Remembering that the three chief virtues are laziness, impatience, and hubris, you can certainly save a lot of time if you look around in the share directory, which contains the source code you seek. I'm not going to give you the details; part of the learning experience is finding out how to look around in (permissible!) directories to find useful stuff.[3]

Our makefile contains the recipe needed to build `proj0`, which is an executable program. You can't simply say

```
g++  -o proj0 proj0.cc
```

because the compiler needs to know about all the header files and libraries needed for the graphical stuff. You'd need to run the command[4]

```
g++ -I/usr/local/include/bjarne/GUI -I -Wall -o proj0 proj0.cc \
-lX11 -ljpeg -lstdc++ -lbookgui -lfltk -lfltk_images
```

Alternatively, you could break it into a compilation step

```
g++ -I/usr/local/include/bjarne/GUI -I -Wall -c -o proj0.o proj0.cc
```

[1] Last semester, I gave you detailed instructions about using the shell to create directories, copy files, and suchlike. Since this is a second-semester course, I expect you to know how to do these things. If you don't remember, go back and look at last semester's handouts; alternatively, look at the Unix/Linux links that are found on the home page of the class website.

[2] The word "makefile" is a generic term for these critters, but `Makefile` is the name of the specific makefile I'm giving you. Calling it `Makefile` rather than `makefile` means that it will appear at the beginning of a directory listing, assuming that the other files in the directory all have names starting with a lower-case letter.

[3] And once you've figured out how to find the source file you need, please don't tell your fellow students; let them figure this out for themselves.

[4] The shell treats a backslash at the end of a line as a continuation character, i.e., it says that the command is too long to fit on one line.

followed by a linking step

```
g++ -o proj0 proj0.o -lX11 -ljpeg -lstdc++ -lbookgui -lfltk -lfltk_images
```

In any case, this is a lot to type. It's easy to get wrong. Typing this several times (if you're building the program incrementally *or* when you have to deal with errors) will be a real chore. The *make* utility comes to your rescue. If you copy the `Makefile` in project's share directory to your working directory, you simply say

```
make
```

and the *make* utility will use the information in `Makefile` to execute the commands that are needed to build the desired target. There's a bit more description near the top of `Makefile`, which tells you about building other targets; you should look at same.

When the command

```
proj0
```

gives the desired graphical image (found on page 431 of the text), then you know that you've successfully completed the assignment.

There are no deliverables associated with this project. If you skip it entirely, your grade will not be directly affected. But if you skip this project and then find (in a later project) that you don't understand the basics about `make` and makefiles and then ask me for help, my first act will be to look at your `proj0` directory—if it's not there, I will then ask you to do Project 0. So save yourself the eventual trouble and do Project 0.

Good luck!!