Assignment 1

Building a Trivial UI with Interface Builder

Application Due: 1:00 pm, Thursday, September 19, 2002

1 Scenario

To show just how multicultural you can be, you've decided to make some Canadian friends. But you've realized that in order to properly fit in with your new friends, you're going to have to face the fact that Canada is *metric*. Even worse, all those cool Canadians you really want to connect with use Macs! To get a sense of how metric units relate to good old imperial units, and to get a feel for Mac OS X, you've decided to write a metric conversion application.

2 Getting to Grips with Mac OS X in the CIS Lab

For this assignment, you will need to use a Macintosh running Mac OS X 10.2, with the developer tools installed. Suitable machines are available in the CIS Macintosh lab, but the instructions for using them are slightly complicated by the needs of other CIS Mac users. If you will not be using the CIS Macs, you can skip the rest of this section.

The key complication for our use of Mac OS X is that

- The CIS Macs boot into Mac OS 9 by default
- The CIS Macs should not be left running Mac OS X

The rationale for this policy is that most CIS users need to use Mac OS 9 for one reason or another, and are likely to be rather confused if faced with an OS X login screen.

2.1 Starting Your Session

If the machine is running Mac OS 9 (very likely, unless you are taking over from another CS 124 student), you will need to reboot into OS X. This process is, sadly, a little slow and tedious because we *do not* use the usual "Startup Disk" method.

Booting Mac OS X from Mac OS 9

1. From the Finder, select Restart from the Special menu.
2. When the screen goes black, hold down the Option key (on the same row as the space bar). Keep holding down the Option key.

3. The Mac will chime. Keep holding the Option key.
4. When the screen goes blue, you can release the Option key.
5. Wait while the boot firmware searches the disks for operating systems to boot. (The mouse pointer will be a watch with rotating hands.) This process takes about one minute, but it usually seems like about ten.
6. Click the OS X icon.
7. Click the right arrow. (After doing so, you should see a grey Apple logo, followed by a boot progress indicator, culminating in a login dialog.)

Logging In

Log in using your `orion` account name and password. Once you have logged in, you will have access to your directory on `odin`. (You can also connect to any files you may wish to access from `kato`, the CIS fileserver, using the Connect to server... menu item of the Finder's Go menu.)

2.2 Ending Your Session

When you are done, you need to log out *and* restart the computer so that it reboots in Mac OS 9. Because the CIS machines boot Mac OS 9 by default, you shouldn't need to stick around while the machine reboots, nor do you need to use the "Option key" boot mechanism.

To log out and reboot, select Restart... from the Apple menu at the top left of the screen. Alternatively, if someone else in CS 124 wants to use the machine after you, select Log Out... from the Apple menu.

3 Getting Started with Mac OS X

If you have primarily used Windows, Linux, or Mac OS 9, some elements of the Mac OS X interface will be new to you. Experiment by doing the following tasks:

- Navigating with the Finder. Determine what each of the the controls on the window title bar do, and what each of the tools on the toolbar do.
- Arranging and using the "dock" (the bar of icons at the bottom of the screen) to your liking. Add and remove applications by dragging. Applications found in `/Applications`, `/Applications/Utilities` and `/Developer/Applications`. You can also add folders to the dock, and contol-click to get various contextual menus.
- Play with some of the Mac OS X applications.

- If you want to look at the Unix side of things, take a look at Terminal (found in `/Applications/Utilities`). As a diversion, you could look at the "transparent terminal window" possibilities available via the Window Settings... menu option of the Terminal menu (i.e., the one next to the Apple).

You may collaborate with other class members in exploring Mac OS X.

4 Create a Simple Conversion Application

To create a simple conversion tool, follow *one* of the following "Cocoa" development tutorials on Mac OS X:

```
/Developer/Documentation/Cocoa/ObjCTutorial/objctutorial.pdf
/Developer/Documentation/Cocoa/JavaTutorial/javatutorial.pdf
```

which are also available on Apple's web site—see the links on the course web page.

The first tutorial shows you how to write a simple Mac OS X application using Objective C. Objective C is a language you probably haven't seen before, but one that should be very easy to learn, especially given the very small amount of code you need to write. Objective C is the native language of Mac OS X user-interface code.

The second tutorial shows you how to write a simple Mac OS X application using Java (using the native UI libraries and tools of Mac OS X). The Java tutorial will allow you to work in a more familiar language, and also includes code examples for a dynamically updating graphic (which you can either skip or implement as you see fit).

Look over both tutorials, and then choose one to follow. Each tutorial differs in what aspects it emphasizes. The Objective C tutorial emphasizes the Model-View-Controller paradigm and spends more time talking about how to lay out controls, which may prove useful in this assignment, but leaves you with an application that converts money (where the exchange rate can vary), whereas the Java tutorial gets you quickly to a temperature-conversion application, but one that may be a little harder to extend.

Don't be perturbed by the 70-page length of the tutorials. You should be able to complete each one in about 90 minutes. There are copies of the tutorials in the CIS Mac lab, and these copies include a couple of corrections (which have also been described in email to the class).

You may collaborate with other class members in getting this part of the assignment done (resolving problems, looking at code, etc.), but what you create must be your own work.

5 Extend Your Converter

Ideally, you should add three of the following features to your converter:

1. A slider allowing a range of values to be explored easily
2. The ability to select different conversions (e.g., Pounds/Kilograms, Miles/Kilometers, Fahrenheit/Celsius, etc.) [Do not use a tab view.]
3. Dynamic text that explains the "rule of thumb" for the particular conversion selected (i.e., explains how to work it out when you don't have the converter)
4. Conversions in both directions (Objective C assignments only)

You may offer coding help to other class members in this part of the assignment. You can help with syntax errors and with looking up documentation. You can only give hints as to what might be useful to look up in the documentation. You may not write code for someone else. You should avoid looking closely at the interface designs of other students in the class—create your own UI design.

Do not spend more than three hours on this part, unless you are having fun.

6 Grading

Grades will be assigned as follows:

A+ Implementing A-grade quality applications using *both* Java *and* Objective C, or some other "far beyond the call of duty" activity

A Implementing all the above features, in an attractive, fully functional application

A– Implementing all the above features in an attractive, mostly functional application

B+ Implementing two of the above features in an attractive, fully functional application

B Implementing two of the above features in an attractive, mostly functional application

B– Implementing at least two of the above features in a mostly functional application

C An implementation of the tutorial code, plus an attempt to go further

D An application that crashes on startup, or unfinished tutorial code

N No attempt at the doing the assignment

In particular, in this class we only care about the look and behavior of your finished application. You do not submit your code, only your compiled application (although you should retain your code and provide it if asked). If your code could cause seizures in CS 70 graders, but works, that is okay. (Of course, having learned the value of clean code in CS 70, you will no doubt write good clean code anyway.)

7 Submission

You will submit your code by copying it to a drop folder on `kato`. The exact procedure will be explained in class and email, but the basics are:

1. Create a new folder in your home directory, and name it so that it matches your CIS `orion` user id. If you wish to resubmit, rename the folder adding `-1`, `-2`, etc. to the end of its name.
2. Copy your application (which can be found in the `build` subdirectory of your project directory) into the new folder.
3. Mount `kato`.
4. Copy your folder into the CS 124 drop folder on `kato`.

The drop folder will not be available until sometime on Monday.