CS 66.460
Introduction to Graphical Human–Machine Interfaces
Spring 1997
Programming Project #1

X Windows Toolkit Programmimg

DUE: FRIDAY, MARCH 7, 1997 5PM

The Goal: To design and implement a program on a campus workstation running **X Windows** which communicates with the user via an interactive graphical user interface.

The Problem: As a software engineer for SuperSoft Incorporated you often have several meetings to attend each week. Recently there has been much confusion at these meetings because more than one group of people are showing up for meetings at the same conference rooms! This is clearly hampering your productivity, since when this happens you spend more time arguing about who is to get the room than in actually carrying out the meeting!

You decide, this is nonsense. I can develop a room scheduler program for everyone to use! That will certainly look good when my performance review comes around next month!

Your new program – which you have dubbed SUPERROOM – will allow users to

- See at a glance which rooms are already reserved and for what dates.
- Reserve a room.
- Delete reservations they have entered.

SUPERROOM will not let anyone schedule a room that has already been reserved. In addition, the system will not allow room reservations to overlap.

What information is needed in order to schedule the room? Here are some suggested parameters:

- the name and phone extension of the person making the reservation;
- the number of people planning to attend the meeting;
- the length (in minutes) of the meeting;
- the type of equipment needed for the meeting;

You are free to use these parameters and/or others as you see fit.

There are only ten conference rooms on-site and each has different characteristics, such as seating and audio-visual hook-ups. All rooms have white boards. The R&D group, your group, is in building 61 upper (61U) but you are free to use the rooms in 61 lower and in building 62 (upper and lower). Someone has already given each room a name:

#	Name	Location	Capacity	O	V	C	M	P
1	Maple	61U	10	O	N	Y	N	N
2	Elm	61U	12	Y	N	Y	N	N
3	Willow	61U	6	Y	N	N	N	N
4	Apple Blossom	61L	4	N	Y	Y	N	N
5	White Pine	61U	8	N	Y	N	N	N
6	Red Pine	62U	10	Y	N	N	N	N
7	Oak	61L	100	Y	Y	Y	Y	Y
8	Maple	62U	15	N	Y	Y	Y	N
9	Mahogany	62U	12	Y	N	N	N	N
10	Palm	62L	50	Y	Y	Y	Y	Y

```
Key:
  O=overhead; V=video; M=microphone; C=computer; P=podium
```

Normal business hours are from 9AM - 5PM, but you know that some people like to schedule meetings earlier and more often later (pizza dinner meetings are your favorite!), so you decide that SuperRoom shall be able to handle room scheduling from 7AM-9PM.

What To Do: Using the above description as a guide, design and implement SuperRoom. Your system should allow users to make reservations and view which rooms are already reserved. But you are encouraged to add additional features and functionality as you see fit. What other features would the user like to have? What other tasks might the user wish to perform?

You may decide to provide different views of the room reservations. Should the user be able to view by day? Week? Month? Should the system automatically suggest a room? Or should the system generate a list of possible rooms and let the user decide?

For this project you are to use one of the X Windows toolkits such as the Motif or Athena toolkits. Other X toolkits may also be used but please obtain approval from the instructor or TA before doing so.

You naturally want to be creative in your design and exploit the features of the underlying toolkit as best you can. Since this is a graphical user interface, you are encouraged to use color and graphics to make the interface visually pleasing and attractive. You need to pay attention to such details as screen layouts, what buttons are available when, and whether to use one big screen or to have several screens or popups or dialogs. You can even use sounds if you wish. As always you should strive to create a user interface that is easy and intuitive to use.

Deliverables: As previously explained, your grade will be primarily determined by a one-on-one *live-demo* of your program with the instructor or the TA. You must schedule this demo at least 24 hours in advance (e-mail preferred). Time slots for demos will be assigned on a first come, first served basis. You should plan on giving your demo during the week after Spring break.

Even though you will not give your demo until after Spring break, you must submit the following **by the due date** in order to avoid a late penalty:

1. The source files used to create your program. This includes all header files, source files, and makefiles you used.
2. A brief user manual which describes what your program does, how to run it, and use it. This may be a plain ASCII file, but make sure it is formatted nicely and that lines do not extend past the 70th column.

You are to submit your source and user manual by e-mailing a uuencoded TAR file to `ghmi-submit@cs.rpi.edu`. This procedure is not yet operational, but when it is, you will be able to send a message with the word `HELP` in the body and receive instructions. Instructions will also be handed out in class or made available on the course Web site.

Remember! This is <u>not</u> a team project. By working alone you will gain the maximum benefit from this project and learn what X Windows toolkit programming is all about. You may consult with friends and classmates on specific issues or seek their help to solve any implementation problems that arise. But you should be writing your own code. Be as creative as you can within the practical limits imposed by the due date and other courses you are taking this semester. Do your best to make your program *robust* so that it will work without crashing no matter what idiotic and unexpected actions the user chooses to perform.