

Name and RCSID:

ECSE-4750 Computer Graphics, Spring 1998
Handout midterm*
Midterm Exam

February 23, 1999

Exam Rules

1. You may have one 2-sided 8.5 inch x11 inch note sheet, which may be mechanically printed. Keep your cribsheet since you can use it again on the final.
2. You may have your blank paper, calculator, pens, etc.
3. You may not communicate with anyone, except Giampiero or me.
4. Answer all questions. Brief, concise, answers are preferred.
5. Spend time on a question proportional to its number of points.
6. Note that the last page is number midterm-10.
7. Start immediately. You have until 1:50.
8. Try to write legibly.
9. Write your NAME on top of this page.
10. Try to write your answers on these question sheets, tho extra paper is allowed. If an answer is on an extra sheet, say so in the normal space on this sheet.
11. Leave the small oval boxes blank; they're for our grade.
12. *Warning:* Be careful of questions that appear to be identical to ones that you've seen before. Something might have changed.

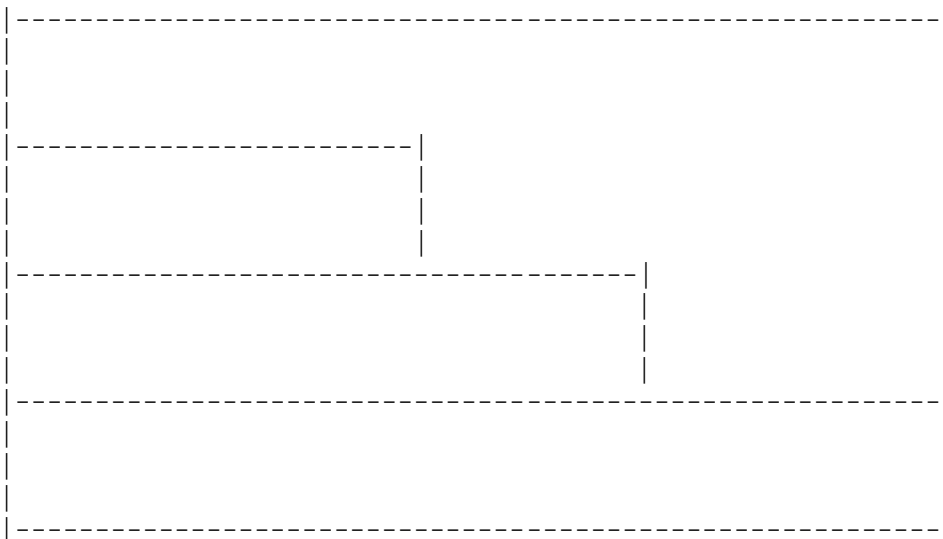
*Copyright 1996-9, Wm. Randolph Franklin. You may copy this for nonprofit research and teaching purposes under these conditions: (1) You do not change the document. (2) You do not charge people to look at or to copy your copy.

Exam

1. /[4] (points) Name four things that `XtVaAppInitialize` does.

2. /[3] Name 3 places where resources can be specified.

3. /[4] In the following diagram, fill in the blanks with the following labels: *motif*, *xlib*, *xt*, *user program*.



4. /[2] Altho Vtcl usually makes laying out a GUI much easier, there is one program that we saw that would be much harder to do with Vtcl, than in Motif or Tcl/Tk. The reason is that in Vtcl, you have lay

out the widgets individually. Circle that program, from the following list, which might extend to the next page if this question is at the bottom of the page.

- (a) xbox.c
- (b) xdraw1.c
- (c) xdraw2.c
- (d) xfarewell.c
- (e) xgoodbye.c
- (f) xhello.c
- (g) xlots.c
- (h) xmove.c
- (i) xtext.c

5. Consider the following X program:

```
/** xexam.c Midterm problem
Time-stamp: </mab_home1/wrf/cg/xexam2.c, Tue, 23 Feb 1999, 19:46:33 EST, wrfmab.ecse.rpi.edu>
*/
```

```
#include <stdio.h>
#include <Xm/Xm.h>                /* Needed by all Motif programs. */
#include <Xm/Label.h>
#include <Xm/PushB.h>
#include <Xm/RowColumn.h>
```

10

```
char    *greek[4] = {"plato","socrates","alex","philip"};
```

```
Widget  widgets[4],  quit,  topLevel;
```

void

```
Proc(Widget w, caddr_t client_data, caddr_t call_data)
{
```

```
    Position x, y;
    int      i;

```

20

```
    i=(int) client_data;
    printf("Client data= %d\n", i);
```

```
    XtVaGetValues(topLevel, XmNx, &x, XmNy, &y, NULL);
```

```
    switch (i) {
```

```
    case 0:
```

```
        x -= 100;
```

30

```
    case 1:
```

```
        x += 100;
```

```
        break;
```

```

        case 2:
            y -= 100;
        case 3:
            y += 100;
            break;
    };
    XtVaSetValues(topLevel, XmNx, x, XmNy, y, NULL);
}

```

40

```

/*
 * quit button callback function
 */
void
Quit(w, client_data, call_data)
    Widget w;
    caddr_t client_data, call_data;
{
}

```

50

```

main(int argc, char **argv)
{
    XtAppContext app_context;
    Widget box;
    int i;

    topLevel = XtVaAppInitialize(&app_context, "XExam",
                                NULL, /* options */
                                0, /* num_options */
                                &argc, /* num cmd line */
                                argv, /* cmd line */
                                NULL, /* fallback */
                                NULL /* args */
                                );

    box = XtVaCreateManagedWidget("box",
                                xmRowColumnWidgetClass, /* class */
                                topLevel, /* parent */
                                NULL); /* args */

    for (i = 0; i < 4; i++) {
        widgets[i] = XtVaCreateManagedWidget(greek[i],
                                xmPushButtonWidgetClass,
                                box, NULL);
    }

    quit = XtVaCreateManagedWidget(
        "quit", /* 12 */

```

70

80

```

                                xmPushButtonWidgetClass,          /* 13 */
                                box,          /* 14 */
                                NULL);      /* 15 */


XtAddCallback(quit, XmNactivateCallback, Quit, 0);


for (i = 0; i < 4; i++)
    XtAddCallback(widgets[i], XmNactivateCallback, (XtPointer) Proc, (XtPointer) i);

XtRealizeWidget(topLevel);

XtAppMainLoop(app_context);
}

```

(a)  [4] What does this program do? Be specific about the action of each button.

(b)  [2] Sketch the tree of widgets from the top level widget to the leaves.

(c) ☐/[2] What are the labels that will print in the command widgets by default?

(d) ☐/[1] What file where in your account will resources be read from (assuming that things are configured as I've described in class)?

(e) ☐/[1] Suppose you want to change the label printed inside (only) the first leaf widget. Give a possible line to add to the resources file.

6. ☐/[1] How can your X program tell the user ran it with command-line arguments that are not X resources and values?

7. ☐/[1] Program `xbox.c` has the following code:

```
/* PopupDialog: callback for the PRESSME button. */
void PopupDialog(w, client_data, call_data)
    Widget w;
    XtPointer client_data, call_data;
{
    Widget parent;
    static Widget dialogbox = (Widget) NULL;
    parent = (Widget) client_data;

    if (!dialogbox) { /* Create this only the first time thru. */
        dialogbox = XmCreatePromptDialog(parent, "dialogbox", NULL, 0);
    }
    XtManageChild(dialogbox);
}
```

How is it that the `if` test creates the dialogbox only the first time thru?

8. ☐/[1] In `xtext.c`, what does the following code accomplish:

```
XtUnmanageChild(XmMessageBoxGetChild(help_widget, XmDIALOG_CANCEL_BUTTON));
```

9. ☐/[1] What is the purpose of a cascade button on a menu bar?

10. ☐/[2] What's wrong with this C program:

```
char command[4]="ls ";
char arg[5]="/tmp";
char line[100];

line=strcat(command,arg);
system(line);
```

11. ☐/[1] Why does X use special data types, like `Point`, instead of just saying `struct { int x,y }`?

12. ☐/[2] According to the *Motif Programming Guide*, chapter 6, "The windows associated with `Pop-upMenus` and `PulldownMenus` are top-level windows. That is, the parent window of such a menu is the

root window of the screen, not the window associated with the parent widget.” What advantage is there to this?

13. /[3] Three types of manager widgets are *rowColumn*, *form*, and *bulletinBoard*. Distinguish between them.

14. Window managers:

- (a) /[2] Name 2 things that a window manager does.

- (b) /[2] If you kill the window manager, what changes in the appearance on the screen? What functionality do you lose?

- (c) /[1] Name any one window manager in general use on RCS.

15. /[1] Consider this code, which creates two widgets.

```
Widget but;  
but = XtVaCreateManagedWidget("but",xmPushButtonWidgetClass,top_level,NULL);  
but = XtVaCreateManagedWidget("but2",xmLabelWidgetClass,top_level,NULL);
```


and this extract from the resource file.

```
*but.labelString: jambo!
```

Does this affect the button widget, the label widget, both, or neither?

16. ☐/[6] For each of the following properties, say whether it applies more to Motif and to Tcl/tk.

- (a) More industry standard.
- (b) Easier to use.
- (c) Faster to execute.
- (d) Funded by ARPA.
- (e) Funded by OSF.
- (f) Source code freely available.

17. Look at the following Tcl program.

```
#!/home/wrf/bin/wish -f

proc getdate { } {
    global date
    set datef [open "|date"]
    set date [read $datef]
    close $datef
}

getdate

button .hello -textvariable date -fg red -command getdate

pack .hello
```

(a) ☐/[1] What does -textvariable do?

(b) ☐/[1] What does -command do?

- (c) ☐ [1] What does the program as a whole do?

Total: 50 points

End of exam

February 23, 1999, 20:10 /dept/ecse/graphics/homidterm.tex