This is a limited and partisan introduction to 'The X Window System', which is widely but improperly known as X-windows, specifically to version 11 ('X11'). The intention of the X-project has been to provide 'tools not rules', which allows their basic system to appear in a very large number of confusing guises. This document assumes that you are using the configuration that I set up at Peyton Hall [†]

There are helpful manual entries under `X` and `Xserver`, as well as for individual utilities such as `xterm`. You may need to add **/usr/princeton/X11/man** to your MANPATH to read the X manpages.

This is the first draft of this document, so I'd be very grateful for any comments or criticisms.

Introduction to X's Anatomy

X consists of three parts:

The server
The part that knows about the hardware and how to draw lines and write characters.

The Clients
Such things as terminal emulators, dvi previewers, and clocks

and

The Window Manager
A programme which handles negotiations between the different clients as they fight for screen space, colours, and sunlight.

Another fundamental X-concept is that of *resources*, which is how X describes anything that a client might want to specify; common examples would be fonts, colours (both foreground and background), and position on the screen.

Keys

X can, and usually does, use a number of special keys. You are familiar with the way that `<shift>a` and `<ctrl>a` are different from `a`; in X this sensitivity extends to things like mouse buttons that you might not normally think of as case-sensitive. In addition, there is another modifier called 'meta' which can be used either alone or in combination with shift or control. On a sun-3 keypad there are two identical meta keys labelled 'Left' and 'Right', on a sun-4 they are labelled 'Alt' and 'Alt Graph'.

[†] if you've lost the files that you were given you can find them in **/u/skel/Userfiles**; you need **.twmrc**, **.xinitrc**, and **.Xdefaults**

Getting Started

To start X, type `xinit`; do *not* type `X` unless you have defined an alias. `X` by itself starts the server but no clients, resulting in an empty screen. (Solution: read the section on 'Problems').

How should you leave X? Go to the black window labelled 'Console' in the top left hand corner of the screen and logout. Why this works will soon become plain.

The X-Server

You shouldn't often need to worry about the server. On our suns we mostly use one called `Xsun`, although it is possible to use one provided by sun called `openwin` (Sun's implementation of X is called OpenWindows; it has an associated window manager called `olwm` and I shall discuss it no further).

It is possible to pass flags to the X-server on the `xinit` command line; for example

```
        xinit -- -zaphod
```

passes the flag `-zaphod` to the server (the '`--`' mean that that arguments should be passed to the server not interpreted by `xinit`).

The Window Manager `twm`

All programmes running under X are equal, but one, the window manager, is more equal. By convention it is appointed and anointed to be responsible for moving windows, drawing one window above another, converting windows into icons and back, and a large number of other tasks. It is the window manager that to a large extent controls the look-and-feel of your screen.

The window manager that you will be using if you use my default startup files is called `twm` (it used to stand for Tom's Window Manager, after the author). When it starts it reads a startup file called **.twmrc** which is described under 'dot-files' later in this document. Most of the description of `twm` that follows assumes that you are using the Peyton Hall default **.twmrc** file (as listed in figure 2 and discussed under 'dot-files').

You will notice that most of the windows on the screen have a title bar at the top (**.twmrc** sets it to a brownish-yellow called `goldenrod`), which has five regions: a square with an X in at the left ($\boxed{X}$), a name, a central region that changes colour when the mouse is in the window, and two squares at the right; one looking like $\boxed{*}$, and one looking like ⧉. If you click a mouse button in the $\boxed{X}$ box the window will be iconised (click in the icon to get it back), if you hold down the left button in the ⧉, a set of lines dividing the window into 9 parts will appear; if you move the mouse off the edge of the window (with the button still down) the outline will follow, and when you let go the window will be resized. If you hold down the left button in the $\boxed{*}$ box and move the mouse the window will be moved.

When you create a new window (*e.g.* by typing `xterm &`) without specifying its position (see the section on 'Command Line Options'), `twm` will put up an outline of the window 'attached' to the mouse. Push Button1 to create it at its current position, hold Button2 down and move the mouse outside the outline to choose some other size, or push Button3 to create the window where it is, but extending to the bottom of the screen.

Instead of icon windows it is possible to use an 'icon manager' that keeps all icons tidily together. You can try this by using the 'window commands' menu (see next paragraph), or choose it permanently by reading the manual.

Table 1 gives the behaviour of the mouse buttons and some other keys. In it the 'modifier' is some combination of nothing, shift, control, or meta, and 'where' is the location where the command works (the root window is the gray background outside any window). 'Button1' is usually the left mouse button,'Button2' is the middle, and 'Button3' is the right. If you want to invert these you can inform the server that you are lefthanded (see `man Xsun`).

Table 1: Key functions defined in the default **.twmrc**

Key	Modifier	Where	Action
Button1	None	Root	Popup the 'machines' Menu
Button2	None	Root	Popup the 'window commands' Menu
Button3	None	Root	Popup the 'utils' Menu
Button1	Meta	Window or Icon	Raise or move† the window
Button2	Meta	Window	Iconise (Close) the Window
Button2	Meta	Icon	Un-Iconise (Open) the Window
Button3	Meta	Window or Icon	Lower or move† the window
Button1		Title‡	Raise or move† the window
Button2		Title‡	Raise/Lower*
Button3		Title‡	Cycle all windows up one
L1		Window	Make a window fill the screen, or shrink it back
L5		Window	Raise/Lower*
L7		Window	Iconise (Close) the Window
L7		Icon	Un-Iconise (Open) the Window
L8			Paste a region into an xterm

† Depending on whether or not you move the mouse while the button is depressed.
‡ 'Title' is the bar at the top of each window. Well, most windows.
* Lower the window if it is fully visible, otherwise raise it to the top.

This discussion applies to `twm`; if you are motivated to explore other managers (such as `tvtwm`) you have progressed before the scope of this introduction.

Command Line Options

Most clients accept a standard set of command line options to choose such things as colours, fonts, and positions. It is very easy to overuse this system; there is a better mechanism (called 'resources') to specify your favourite options. You should probably only use command line options to override your usual choices — for example the 'Console' window created in my **.xinitrc** file is the only one that I want to be black.

The more useful flags are:

-bg colour	Set the background colour
-fg colour	Set the foreground colour
-fn name	Choose a font
-geom 100x200+30-50	Set the window's geometry; see below
-title string	Set the window's title

X understands symbolic names for colours. If you don't want to settle for red, green, or yellow then you can use `xcolors` to choose `tomato`, `aquamarine`, or `gold` instead. Rumour has it that the names come from a very large box of crayola crayons sometime back in the history of X or its predecessor W.

Font names can be very simply or horribly complex. I usually use forms like `9x15`, but I could just as well say

```
-misc-fixed-medium-r-normal--15-140-75-75-c-90-iso8859-1
```

There is a utility called `xfontsel` that might help you decide.

Geometry strings look strange, but are simple. The first 2 numbers, if present, specify the size of the window (in this case 100×200). Most clients interpret the size in pixels, but some interpret them in more personal ways; for example terminal emulators often interpret the size as being in characters.

The final two numbers are the position of the window and may be omitted. When a value is positive it applies to the top or left corner relative to the top or left of the screen; when it is negative it's the bottom or right relative to the bottom or right of the screen. Hence, `+100-1` means that the window should be 100 pixels from the left, and touching the bottom of the screen.

Resources

The default resources file **.Xdefault** is given in Figure 3. It consists of a number of lines, each of which sets some property of either a given client, a class of clients, or of a class of parts of clients. The defaults file is read and installed by the client `xrdb`.

The syntax of resources is described in the `X` man page, but briefly each line consists of a property with a number of fields seperated by '.' or '`*`', then a colon, then the corresponding value. If the property begins with the name of a client it only refers to that programme, so

```
xclock.geometry:  90x90-1+1
```

specifies a geometry string for `xclock`, while

```
*Font:  9x15
```

says that the font should always be `9x15` unless otherwise specified. An example of a resource associated with a part of many clients would be

```
*Command.ShapeStyle:  Oval
```

which specifies that all command buttons should be oval rather than square. In the provided file it is commented out with an '!' in the first column.

Clients

This is a very large topic, as there are a very large number of programmes available, ranging from games through terminal emulators to such well-loved masterpieces as `SM`. Some programmes, such as emacs, are not fully-fledged X applications, but make use of some exy features such as moving the cursor with the mouse. A number of the more popular programmes around Peyton Hall are:

ghostview Preview postscript-files.

xbiff (or `xpbiff`) Notify you when mail arrives. It isn't entirely clear that this is a good idea, as it makes email only slightly less intrusive than the telephone.

xclock Put up an analogue or digital clock.

xcalc Create a calculator, either RPN or some other sort.

xdbx An X-based front-end to the debugger `dbx`. I'd probably recommend using gdb `instead`

xdvi Preview files produced by TEX.

xxgdb An X-based front end to gdb.

xhost Control access to your screen from other machines.

xinfo Read info files (which are the documentation for a good many of the GNU programmes like emacs, as well as some local ones such as mirella and SM).

xman Browse manual pages.

xphoon Display the moon, with the proper phase, in your root window.

xterm You've already met this one; it's the basic terminal emulator, capable of impersonating either a vt100 or a tek4014. One thing that's pretty common is cutting text (so as to paste it somewhere else). There are many ways to do this, but basically hold down the left button and move it around; a highlighted region appears. Once you let go the region can be extended with the right button. To paste it into a different `xterm` window use the centre button or the L8 (Paste) key. It is also possible to paste into other consenting clients, see the man pages or some other source of documentation for details.

Possible Problems

1. You started the server without starting any clients.

This usually happens if you think that `X` is an alias for `xinit`, but it isn't. Go to another terminal, find `X`'s process ID (`ps | grep X`) and kill it (`kill 12345`). If you say `kill`

`-9 12345` you will be sorry as the console will appear hopelessly confused. Return to your other terminal, say `kbd_mode -a`, and make a note not to use `-9` without due reason.

2. `X` dies, and none of the keys make any sense.

   See item 1 for a fix.

3. The mouse seems to have disappeared, or maybe it hardly responds to your frantic hand signals.

   This probably means that the hardware in your workstation has support for more than one screen, and you have accidently moved the mouse onto one that you can't see. You can usually get the mouse back by moving it off various edges of the screen; you can prevent the problem happening by passing the '-zaphod' flag to the server: `xinit -- -zaphod`. Someone should be able to explain the etymology to you.

'Dot'-files

.xinitrc

When you say `xinit` the X server is started, and the commands in the file **.xinitrc** are executed. Figure 1 (at the end of this document) shows the **.xinitrc** that I provide for new users at Peyton Hall. Let us take a quick tour through this file, as it is where you add commands to taylor your display to your personality.

The environment variable `DISPLAY` tells X where to put its windows; for example if DISPLAY were `grendel.princeton.edu:0` then they would appear on my machine. You would be correct in thinking that this implies that X is able to operate over a network. The ':0' means grendel's first display; if grendel had two displays the second screen would be `grendel:1`.

Because it would be irritating if anyone could put up a window on my workstation, X normally only accepts windows from the machine that the server is running on; the `xhost` command allows access from other machines (in this case from `astro`).

Resources next: the line

```
xrdb -load $HOME/.Xdefaults
```

means read the ResourceDataBase from the file **.Xdefaults** in your home directory.

The window manager `twm` is started next, followed by some clients (a terminal emulator, a load-average monitor, and a clock), and then a terminal emulator (`xterm`) that functions as a console. Note that there is no `&` at the end of this line, so **.xinitrc** stops and waits for this `xterm` to finish. When it *does* finish, the remaining commands cleanup a bit, the script comes to a conclusion, and the X server is terminated. This means that logging out of the console window kills X, as claimed above.

.twmrc

When `twm` is started from your **.xinitrc** file it reads its own startup file, called **.twmrc**; see Figure 2 for an example. For full details you should of course refer to the manual page (`man twm`).

The file starts by setting a couple of variables (`NoGrabServer` and `NoTitleFocus`), then it defines colours for various purposes, chooses a font, specifies a region to 'park' icons in, and then after some magic proceeds to bind commands to mouse keys.

The line

```
Button1 = :  root :  f.menu "machines"
```

means that mouse button one, if pushed in the root window, will call the function `f.menu` with argument `machines`. Equally,

```
Button1 = m :  window|icon :  f.function "move-or-raise"
```

means that button one, if pushed when the 'meta' key is down, in a window or an icon, will call the function 'move-or-raise'. I leave it to your studies of the manual to figure out how this is defined.

After defining some of the keys at the left of the keyboard to correspond to their labels, the file proceeds to define the three menus that you know how to popup from the root window. The syntax should be self evident.

Figure 1: a sample **.xinitrc** file

```
#!/bin/sh
DISPLAY=unix:0 ; export DISPLAY
#
xhosts="astro"                          # Machines with access to your screen
for f in $xhosts; do
        xhost +$f 2>&1 > /dev/null
done
#
XUSERFILESEARCHPATH=/peyton/lib/X11/app-defaults/%L/%N
export XUSERFILESEARCHPATH
xrdb -load $HOME/.Xdefaults
#
twm &                                   # window managers
xterm -geometry 80x48+1+103 &
xclock &
#
xnetload -geom 240x75+815+1 astro &
#
# Console window is last -- logout to kill X
#
xterm -geometry 80x5+1+1 -C -fg wheat -bg black -T Console
#
# Cleanup
#
for f in $xhosts; do
        xhost -$f 2>&1 > /dev/null
done
kbd_mode -a
```

Figure 2: a sample **.twmrc** file

```
NoGrabServer
NoTitleFocus

Color
{
        BorderColor "slategrey"
        TitleBackground "goldenrod"
        TitleForeground "black"
        MenuBackground "aquamarine"
        MenuForeground "black"
        MenuTitleBackground "white"
        MenuTitleForeground "goldenrod"
        IconBackground "white"
        IconForeground "black"
        IconBorderColor "slategrey"
}

IconFont        "9x15"

IconRegion "=216x90+600+1" north east 0 0

#NoMenuShadows

NoTitle {
        "xclock"
        "xnetload"
}

RestartPreviousState
#
# Define some useful functions for motion-based actions.
#
MoveDelta 3
Function "move-or-lower" { f.move f.deltastop f.lower }
Function "move-or-raise" { f.move f.deltastop f.raise }
#
# Set some useful bindings.
#
RightTitleButton "star" = f.move
Button1 = : root : f.menu "machines"
Button2 = : root : f.menu "window commands"
Button3 = : root : f.menu "utils"
#
Button1 = m : window|icon : f.function "move-or-raise"
Button2 = m : window|icon : f.iconify
Button3 = m : window|icon : f.function "move-or-lower"

Button1 = : title : f.function "move-or-raise"
Button2 = : title : f.raiselower
Button3 = : title : f.circleup
#
# These are L5 ("Front") and L7 ("Open") on a sun keyboard
#
"F11" =         : all : f.fullzoom              # L1/F11
"F15" =         : all : f.raiselower            # L5
"F17" =         : all : f.iconify               # L7
```

Figure 2: a sample **.twmrc** file (continued)

```
#
# Define menus:
#
menu "machines"
{
"Machines"      f.title
"Host"          !"xterm -geometry 80x24 &"
"Airy"          !"xterm -geometry 80x24 -T xterm@airy -e rsh airy&"
"Astro"         !"xterm -geometry 80x24 -T xterm@astro -e rsh astro&"
"Bessel"        !"xterm -geometry 80x24 -T xterm@bessel -e rsh bessel&"
"Betelgeuse"    !"xterm -geometry 80x24 -T xterm@betelgeuse -e rsh betelgeuse&"
#"Grendel"      !"xterm -geometry 80x24 -T xterm@grendel -e rsh grendel&"
"Mira"          !"xterm -geometry 80x24 -T xterm@mira -e rsh mira&"
"Tantalus"      !"xterm -geometry 80x24 -T xterm@tantalus -e rsh tantalus&"
}

menu "window commands"
{
"Twm"   f.title
"Restart twm"   f.restart
""              f.nop
"Refresh"       f.refresh
"Resize"        f.resize
"Show Iconmgr"  f.showiconmgr
"Hide Iconmgr"  f.hideiconmgr
""              f.nop
"Kill"          f.destroy
"Delete"        f.delete
}

menu "utils"
{
" Utils " f.title
"Emacs" !"emacs -geometry 80x51-1+112 &"
"Lock"  !"xlock -mode life -nice 8 &"
}
```

Figure 3: a sample **.Xdefaults** file

```
*BorderWidth:                   3
*BitmapIcon:                    off
*Background:                    moccasin
*JumpScroll:                    1
*Font:                          9x15
!*Command.ShapeStyle:           Oval
xclock.clock.background:        navy blue
xclock.clock.chime:             false
xclock.clock.Foreground:        yellow
xclock.geometry:                90x90-1+1
xclock.clock.update:            1
xdvi.densityPercent:            25
xdvi.geometry:                  +1+1
xdvi.pixelsPerInch:             270
xdvi.topMargin:                 0.9
xterm*Font:                     9x15
xterm*cursorColor:              black
xterm*Foreground:               black
xterm*InternalBorder:           5
xterm*pointerColor:             blue
xterm*scrollBar:                true
xterm*scrollKey:                true
xterm*VT100.Translations:       #override \
                Shift <Btn2Up>: insert-selection(CUT_BUFFER0)\n\
                <Key>F18:       insert-selection(CUT_BUFFER0)\n
```