GUI Programming
CS3283

Hugh Anderson

Contact information

Room	S15 #06-12
Telephone	6874-6903
E-mail	hugh@comp.nus.edu.sg

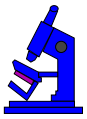
BAD and GOOD

GUI definition

The use of pictures rather than just words to represent the input and output of a program.

A program with a GUI runs under some windowing system.

Original idea from research at SRI by Doug Engelbart.



GUI

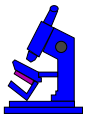
Icons, buttons, dialogue/dialog boxes, windows on the screen.

User controls by moving a pointer on the screen and selecting objects.

Though Apple Computer would like to claim they invented the GUI with their Macintosh operating system, the concept originated in the early 1970s at Xerox's PARC laboratory.

Official description of CS3283

This module aims to teach the nuts and bolts of GUI programming. At the end of the course, students will acquire practical knowledge in Windows programming and techniques of programming interactive systems. Topics include Windows programming, Motif, Tcl/Tk programming.

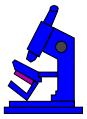


And...

- ✔ Graphical visualization

My expectation

- ✔ Read supporting papers, and
- ✔ maintain an active interest in GUI design and implementation.



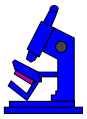
Assessment

Assessment		Weighting	Grade
Assignments			35%
Ass1	Group	10	
Ass2	Individual	20	
Ass3	Individual	40	
Ass4	Group	30	
Tutorials			5%
Mid-term	Closed book		10%
Final Exam	Open Book		50%
Total marks			**100%**

Readings...

1. My textbook, and

1. The User Interface Concepts & Design, Lon Barfield, Addison-Wesley (1993)
2. The JFC Swing Tutorial - A Guide to Constructing GUIs, Kathy Walrath & Mary Campione, Addison-Wesley (1999)
3. Tcl and the Tk Toolkit, John K. Ousterhout, Addison-Wesley (1994)



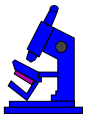
Coverage

- Fundamental GUI concepts (1 lecture)
- Design and programming techniques (3 lectures)
- Cross platform GUI development (6 lectures)
- Visualization techniques (2 lectures)

Enjoy the course!

Chapter 1

Module 1 - GUI concepts

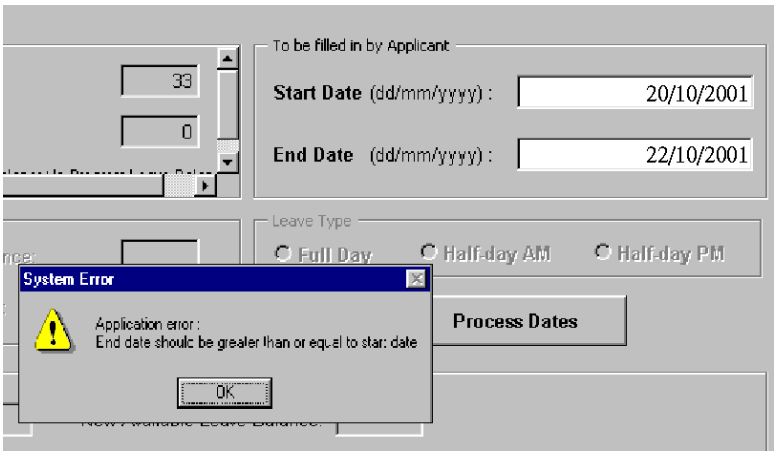


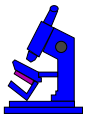
GUI concepts

- ✔ Early user interfaces text based - fixed event ordering.
- ✔ GUI provides for complex interaction, and
- ✔ GUI relies on shared concepts or metaphors.

GUI programming is about the conceptualization, design and implementation of that part of a software application which is concerned with user interaction.

How not to do GUI

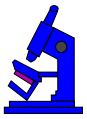




Two points

1. Try out your applications before delivering them.
2. Ensure that error messages are precise, and indicate the next step.

How not to do GUI



General rules of GUI

Effective GUIs owe more to effective psychology than to effective programming.

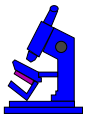
✔ Not just icons - includes an abstract view

✔ No clash between views

EZ-link

Clash between

- User view: pay afterwards
- Bus co view: pay first



Key points

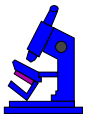
Another key point is that humans are not equipped to handle multiple things at one time, and this leads us to try to keep interfaces simple and uncluttered.

Humans are particularly good at navigating systems which have some analogy to things they know - for example the use of the desktop metaphor is well established and works well in most cultures. Icons are also useful, but shouldn't be abused.

Summary

1. Ensure correlation between What-u-c and What-u-think
2. KISS
3. Analogy, metaphor and icons

Always remember to include the U in GUI.



Do's and don'ts

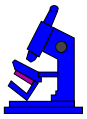
- **Do follow standards**
- **Do be predictable and responsive**
- **Do be flexible**
- **Don't forget the user**
- **Don't forget the machine/environment**
- **Dont assume things**

Types of applications

Not all applications benefit from a GUI - consider embedded systems

However, there are areas that do benefit from a GUI:

- **Immersive applications**
- **Offi ce and business applications**
- **Interactive control systems**



And visualization ...

In addition, a newer application area involves the use of visualization to examine large data sets:

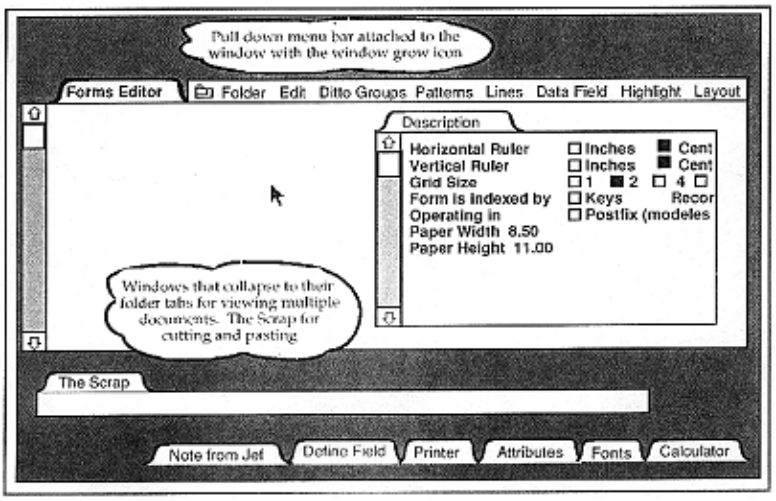
- **Data mining:** - delving into some set of data.

And WAP, PDA ...

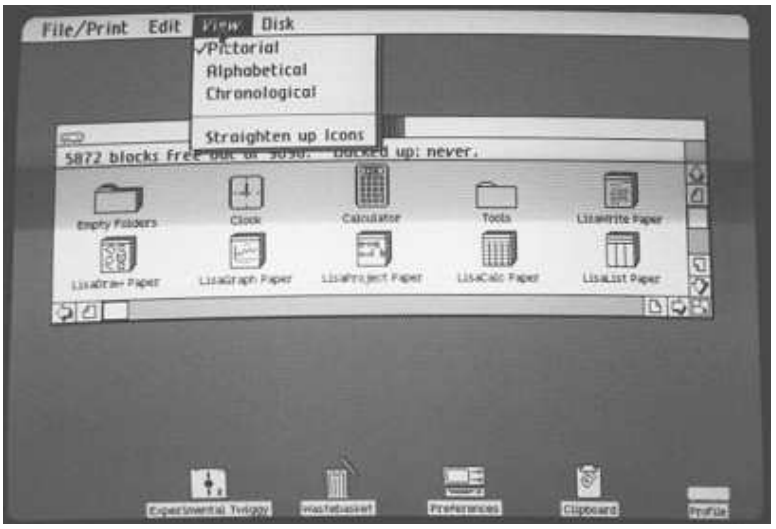
Finally, there is the use of GUI in WAP enabled devices and on PDAs.

This is a specialist topic, which will not be covered in this course.

MAC platform



MAC platform



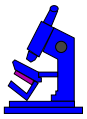
MAC platform

X

The X window system[1] is a sophisticated and well developed system which allows for multimedia software and hardware components to be distributed around a network.

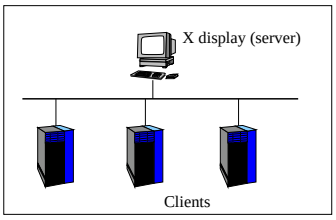
✔ At its simplest level, it allows a program and its display to be on different computers.

[1] The system is called X, or the X window system. UNIX weenies insist that it is **not** called X-windows!



X architecture

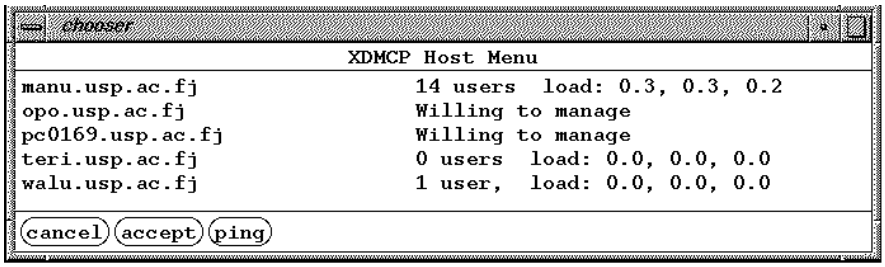
The architectural view of X is a little peculiar. The designers view the display as central to the system, and the software running on the display is called the X-server:



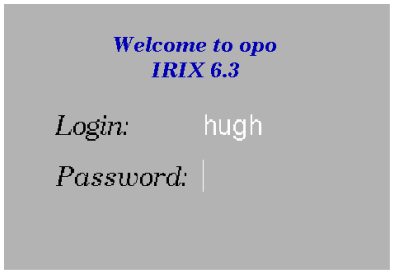
Components of X

1. **The X server**
2. **The X protocol**
3. **X clients**
4. **The Window manager(s)**
5. **The Display manager(s)**

Display manager

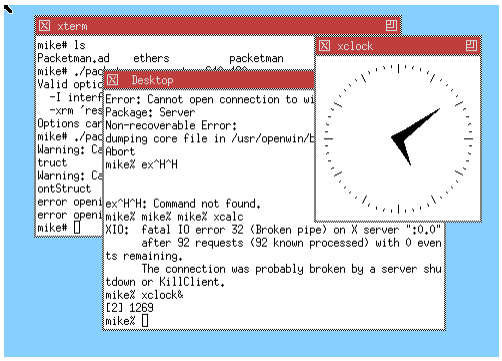


Display manager

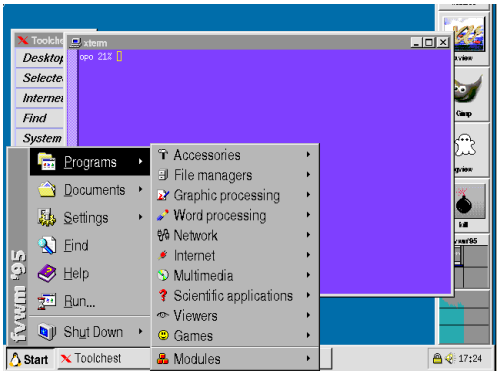


or

Window managers

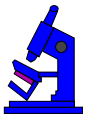


or



Win32

- ✔ Win32 is the 32 bit successor of the Win16 API
- ✔ Win32 is a generic name for 4 (slightly) different APIs
- ✔ The Win32 API on Win95 is a subset of those on WinNT, so applications written for Win95 should be portable to WinNT.



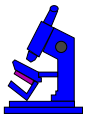
Using Win32

- ✔ The normal way for you to access Win32 functions is by using a precompiled library from a C program.
- ✔ C programmers include a set of header files, and applications link at run time to the Win32 DLLs.

API for Win9X

Has three sections:

- **KERNEL:** - the low level kernel services in kernel32.dll.
- **GDI:** - Graphics Device Interface - drawing and printing in gdi32.dll.
- **USER:** - User Interface controls, windows and messaging services in user32.dll.



Non-native platforms

The following systems can be used to provide a consistent environment that is independant of the host operating systems:

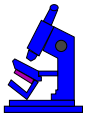
- Java/Swing
- Web browser interfaces
- Thin client systems

Java

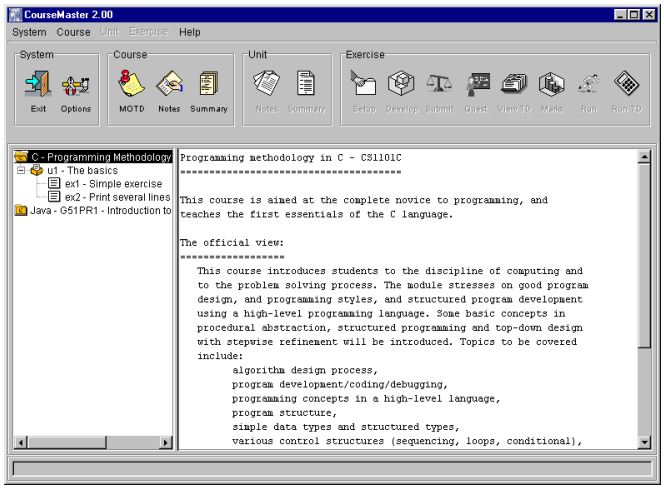
Sun Microsystem's development of Java has always been done with portability issues in mind.

It is relatively easy to write a portable application - for delivery either as an applet in a web page, or as a standalone application.

The **swing** windowing toolkit is the Java API for GUI development.

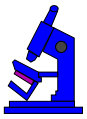


Java app



Web browser interfaces

- ✔ The first web servers provided static pages of hypertext and images.
- ✔ Demand led to the specification of a standard for active page generation - CGI

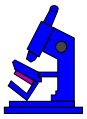


CGI

- ✔ CGI specifies how to pass arguments to a program on a server as part of the HTTP request.
- ✔ The program might then look up a database before generating some HTML to pass back to the browser.
- ✔ A CGI program can be any program which can accept command line arguments
- ✔ Perl is a common choice for writing these programs.

Warning

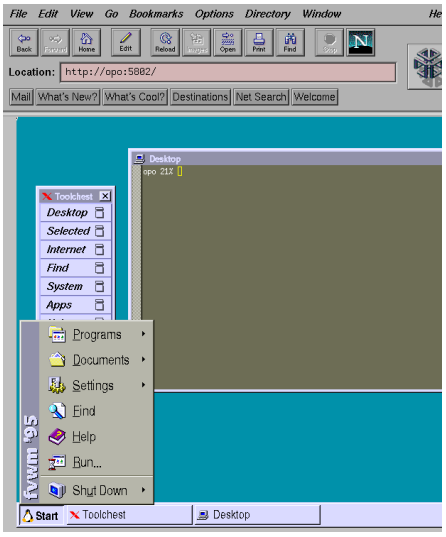
- ✘ You should be aware that poorly constructed CGI scripts can result in security problems for the server, and
- ✘ there is normally a process overhead for each script started.

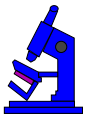


Other web interfaces

1. Java applets, to allow processing at the browser,
2. PHP (a server-side, cross-platform, HTML-embedded scripting language), or
3. ASP (a scripting environment for Microsoft Internet Information Server in which you can combine HTML, scripts and reusable ActiveX server components).

Thin client systems



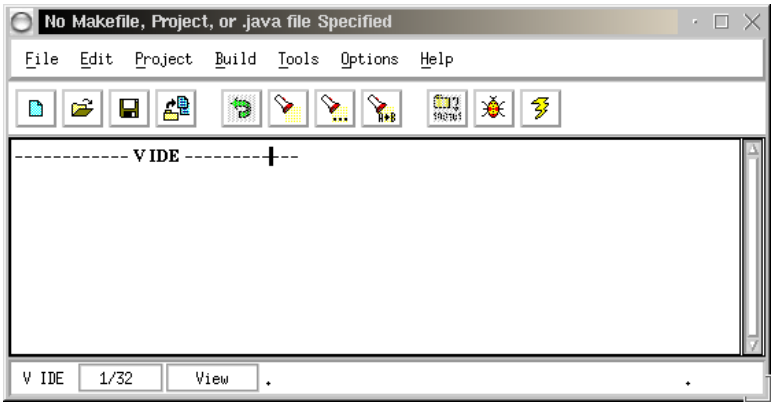


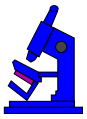
Widget sets

[possibly evoking "window gadget"] In graphical user interfaces, a combination of a graphic symbol and some program code to perform a specific function. E.g. a scroll-bar or button. Windowing systems usually provide widget libraries (sets) containing commonly used widgets drawn in a certain style and with consistent behaviour.

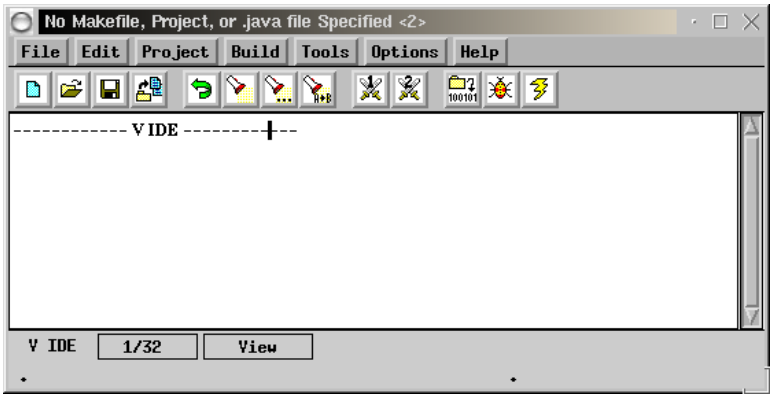
When we use different widget sets, our applications have a slightly different look-and-feel.

Motif application



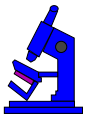


Athena application



Widgets as components

The ICS widget databook has a series of useful widgets to extend the basic Motif set, including ones for bar graphs and so on.



Summary of topics

In this module, we introduced the following topics:

- Rules of GUI
- Types of applications
- Windowing/GUI environments
- Widgets