Section I

GUI Concepts
Developers Point of View

★ a means by which an application program can obtain input from and display output to a user of the application

★ provides the intermediary mechanisms for communication between the application and the user

★ these mechanisms appear as a set of objects with graphical representations on the screen

★ different object represent different means of attaining input or providing output
GUI Concepts

★ emphasis on improving the ease of use of a system

★ aim of a GUI system is to enable the user to: **perform more by doing less**
else the GUI is useless

★ developing GUIs must take very little effort and time in order to focus on the
more essential parts of the application

★ usually an API is provided in order to provide access to the GUI framework
Types of GUIs

★ character-based GUIs
  – do not require the X Window System
  – text-based terminal are able to execute these applications
  – improve interaction over regular command line

★ framebuffer-based GUIs
  – requires the X Window System
  – runs on top of a desktop environment or window manager
  – improve interaction by enabling the use of pointing devices and multiple windows
Window

- area in which your application resides
- a collection of widgets
- differentiate between your application and other application on the screen
- contains a number of basic elements:
  - minimize/maximize buttons
  - close button
  - resize
  - window title
- each window in an X Window System contains a unique ID number
- windows can be reference by these numbers
Widgets

★ basic elements of a GUI

★ each graphical user interface environment contains these

★ some common widgets are:
  – labels
  – scrollbar
  – text box
  – button
  – toggle button
  – ... many others ...

★ certain widgets are able to trigger events
Layout

★ special widgets that contain other widgets
★ created to enable developers to easily layout widgets in a window
★ some common layouts are:
  – grid bag
  – bulletin board
  – row column
  – free
★ layout widgets can be placed atop other layout widgets
Events and Event Handlers

★ program user reaction is determined by a series of **Events**

★ are usually trigger by input devices and widgets

★ also called **Actions** or **Signals**

★ after an **event** occurs an function is called

★ these functions are called **event handlers**, **action handlers** or **signal handlers**
Section II

GUI Development Toolkits
Toolkit Goals

★ widgets are to hold data and present an interface to the user

★ management of widget geometry

★ dispatching and handling of events

★ more features are already toolkit dependent
Toolkits

★ ncurses
★ tcl/tk
★ Xt
★ Motif
★ GTK+
★ Qt
ncurses

☆ new curses

☆ a replacement to the old text based screen handling GUI API called curses

☆ created to abstract the developer manually handling the details of different terminal types

☆ used to create character-based GUIs

☆ provides a C API

☆ the development libraries are released under an open source license
tcl/tk

★ Tcl (Tool Command Language)
  – is simple to use command language
  – it has a simple and programmable syntax and can be either used as a standalone application or embedded in application
  – is open source software

★ Tk
  – is a graphical user interface toolkit
  – makes it possible to create powerful GUIs incredibly quickly
Xt

★ X Toolkit Intrinsics
★ a layer above the Xlib interface
★ provides a function set of widgets
★ provides a C API
★ too cumbersome to use
★ creating a simple application needs a significant amount of code
★ the development libraries (XFree86) are released under an open source license
Motif

☆ probably the most popular Unix GUI toolkit

☆ provides an added layer of abstraction from Xt

☆ used to create CDE and CDE applications

☆ numerous GUI applications are written in Motif

☆ provides a C API

☆ the development libraries are released under an open source license

☆ **Motif builder** is a commercial WYSIWYG tool for building Motif applications
GTK+

★ GIMP (General Image Manipulation Program) Toolkit
★ initially intended to support the development of GIMP
★ is a multi-platform C toolkit for creating graphical user interfaces, primarily designed for the X Window System
★ composed of three essential components:
  – **GLib** - provides many useful data types, macros, type conversions, string utilities and a lexical scanner
  – **GDK** - wrapper for low-level windowing functions
  – **GTK** - advanced widget set
★ used by the GNOME project
★ the development libraries are released under an open source license
★ **glade** is a open source WYSIWYG tool for building GTK+ applications
Qt

★ is a cross-platform C++ toolkit for application development

★ provides a platform-independent API to all central platform functionality:
  – GUI
  – database access
  – networking
  – file handling

★ provides a C++ API

★ used by the KDE project

★ the development libraries are released under an open source license

★ qt-designer is a open source WYSIWYG tool for building GTK+ applications
Copyright © 2000-2001 by William Emmanuel S. Yu. This material may be distributed only subject to the terms and conditions set forth in the Open Content License, v1.0 or later (the latest version is presently available at http://opencontent.org/opl.shtml).