CS 160: Lecture 11

Professor John Canny
Spring 2003
March 10
Finish windows and events

Platforms:
* PC (Viz Basic, Java)
* Web (HTML, XML, Javascript)
* Windows CE
* BREW
* Services - XQuery
Event-Driven Programming

- All generated events go to a single event queue
  * provided by operating system
  * ensures that events are handled in the order they occurred
  * hides specifics of input from apps
Widgets

- Reusable interactive objects
- Handle certain events
  * widgets say what events they are interested in
  * event queue/interactor tree sends events to the “right” widget
- Update appearance
  * e.g. button up / button down
Widgets (cont.)

Generate some new events
* “button pressed”
* “window closing”
* “text changed”

But these events are sent to interested listeners instead
* custom code goes there
Main Event Loop

while (app is running) {
    get next event
    send event to right widget
}

Mouse Software

Keyboard Software
Interactor Tree

- Decompose interactive objects into a tree
  - Interactive objects also known as “widgets”
  - Based on screen geometry of objects
  - Nested rectangles
- Used for dispatching events
  - Events are dispatched (sent) to code in widget
  - The code then handles the event
- Variety of methods for dispatching events
  - Return to this later
Interactor Tree 1

Display Screen

- "F:\cs160\Public" window
  - Inner Window
    - title bar
    - horizontal scroll bar
    - contents area
      - "CDJukebox" folder
      - "Home Ent..." folder
      - ...
    - size control
    - ...
  - "Web Newspaper" window
  - ...

3/10/2003
Interactor Tree 2

Display Screen

- Outer Win [black]
- ??????
Interactor Tree 2

Display Screen
- Outer Win [black]
- Inner Win [green]

Result Win [tan]
- Result String

Keypad [Teal]
- = button
- - button
- + button
- 0 button

93.54

3/10/2003
Interactor Tree (Java)

Display Screen
- Frame [black]
- Panel [green]

Text Entry [tan]
- Result String

Keypad Panel [Teal]
- Button(“=”)
- Button(“-”)
- Button(“+”)
- Button(“0”)

93.54
Who gets the events?

To catch events, a widget registers a "listener" for that event

* Mouse click
* typed input
* drag...

Events go to a widget containing the pointer
Who gets the events?

- But there are often several widgets containing the pointer

- Events go down the “stack” of visible widgets at the pointer until there is a widget that has registered a listener for that event
Interactor Tree (Java)

Display Screen
- Frame [black]
- Panel [green]

Text Entry [tan]
- Result String

Keypad Panel [Teal]
- Button(“=”)
- Button(“-”)
- Button(“+”)
- Button(“0”)

Mouse click listener

3/10/2003
Interactor Tree (Java)

If a widget isn’t visible at the mouse, you may have to “redispatch” events to it. See http://sun.java.com/javaone/track1.html#awt.tut
Platforms - PC

- For regular PC development, the options are:
  - C++/VBasic (Visual Studio)
  - Java
  - Rapid prototyping: Suede, Silk, Satin (see guir.berkeley.edu/projects)
Platforms - Web

- For web development one of the main issues is portability. Before designing your app, think about browsers for your user group.
- There is a lot more than IE and Netscape:
  - Mozilla/Opera
  - AOL: huge community, many versions with limited browsers
  - Old versions of IE and Netscape
Web standards

- Unfortunately, HTTP is a non-standard. The current version is HTML 4 (1997), but no browsers fully support it.

- Microsoft seems to have given up on HTML 4 in 1998.

- Reasonable support for HTML 4 in Netscape 7 and Mozilla. (but tables are different in most browsers)
Web standards

- For portability, it’s best to stay with HTML 3.2

- Javascript is the most portable script. But you’ll probably still need browser-specific code (e.g. CS160 lecture page).
Web standards - XML

- Fortunately, the situation looks better in future. XML is going to become the standard for web info exchange.

- XML provides data exchange, and complementary standards control formatting – XSL and XHTML.

- Good support in Mozilla, also IE and Netscape.
XML Graphics standards

There are two standards for 2D graphics:

- VML (old) promoted by Microsoft - static 2D graphics, available in MS IE now. (example)

- SVG (new) dynamic 2D graphics, the latest W3C standard. No browser support natively, but plug-in available (Adobe), and custom builds of Mozilla support it.
PDAs

Two options for native development–
- MS Embedded Visual Tools 3.0 (VB and C++) – includes emulators for all platforms (download).
- MS Visual Studio .NET (huge!) includes tools for XML exchange. - Ask william@eecs for CDs

Java: Chai VM for HP Jornadas etc.
- Usually well behind PC Java - no Jini support

Flash: Interesting choice for small devices, better use of limited screen space, but check functionality (esp. script execution).
BREW is Qualcomm’s “Binary Runtime Environment for Wireless”

Something like the WIN32 API, but smaller. BREW 2.0 includes support for

- GPS-one - though no providers yet
- Can get GPS info through serial port on emulator PC.
When prototyping mobile apps, the backend services play a large role.

Since XML is the lingua franca for inter-device communication, a general-purpose XML database engine is ideal.

Cerisent XQE server for CS160 running at the URL given in class.
XML services - Cerisent XQE

- Background reading on XQuery: