

Graphical User Interface

Larry Rudolph

MIT 6.893; SMA 5508

Spring 2004






User Interface

- Application needs external interaction.
 - Disconnect application from user interface
 - UI could be: character display & keyboard
 - Or: file, tape, CD, ...
- Graphical user interface
 - Must make many assumptions about h/w
 - Code usually written to lowest common denominator
 - Why write different code for each fancy featyre
 - Code gets tied to GUI
 - Abstraction advantage is minimized
 - hard to adapt to different interfaces
 - e.g. try to access web page via dumb terminal (lynx)





Signals and Actions

- User Interface
 - output: data & signals (event == signal)
 - data is usually text to display to user
 - cause something to happen e.g. popup window
 - input: data & signal
 - data is usually text that the user types
 - event is other user action, e.g. enter key, mouse click, mouse movement
 - Two types of signals (events)
 - window manager, e.g. kill window, resize
 - application UI, e.g. menu selection
 - Signals invoke application code
 - “call back” or “handler”
- 



Evolution of GUI

- Evolution
 - application code does everything
 - libraries provide uniformity
 - apps examine all events & pass on irrelevant ones
 - multi-tasking pushes routing down to OS level
 - window manager (pull routing out of OS)
 - multi-threading: data & signals passed between app and gui via window manager
 - full separation: xml, soap, ...



Graphical Toolkit

- Need something to draw graphics
- Need something to manage windows, menus, buttons, icons, etc.
- Need a “Toolkit”
 - There are lots of them; some language specific
 - We use GTK 2 (pygtk2 python interface)

<http://www.moeraki.com/pygtkreference/pygtk2reference/>

<http://www.async.com.br/faq/pygtk/index.py?req=index>





Graphical User Interface

- We will use Glade 2 : can see what gui will look like (but maybe not on ipaq small screen)

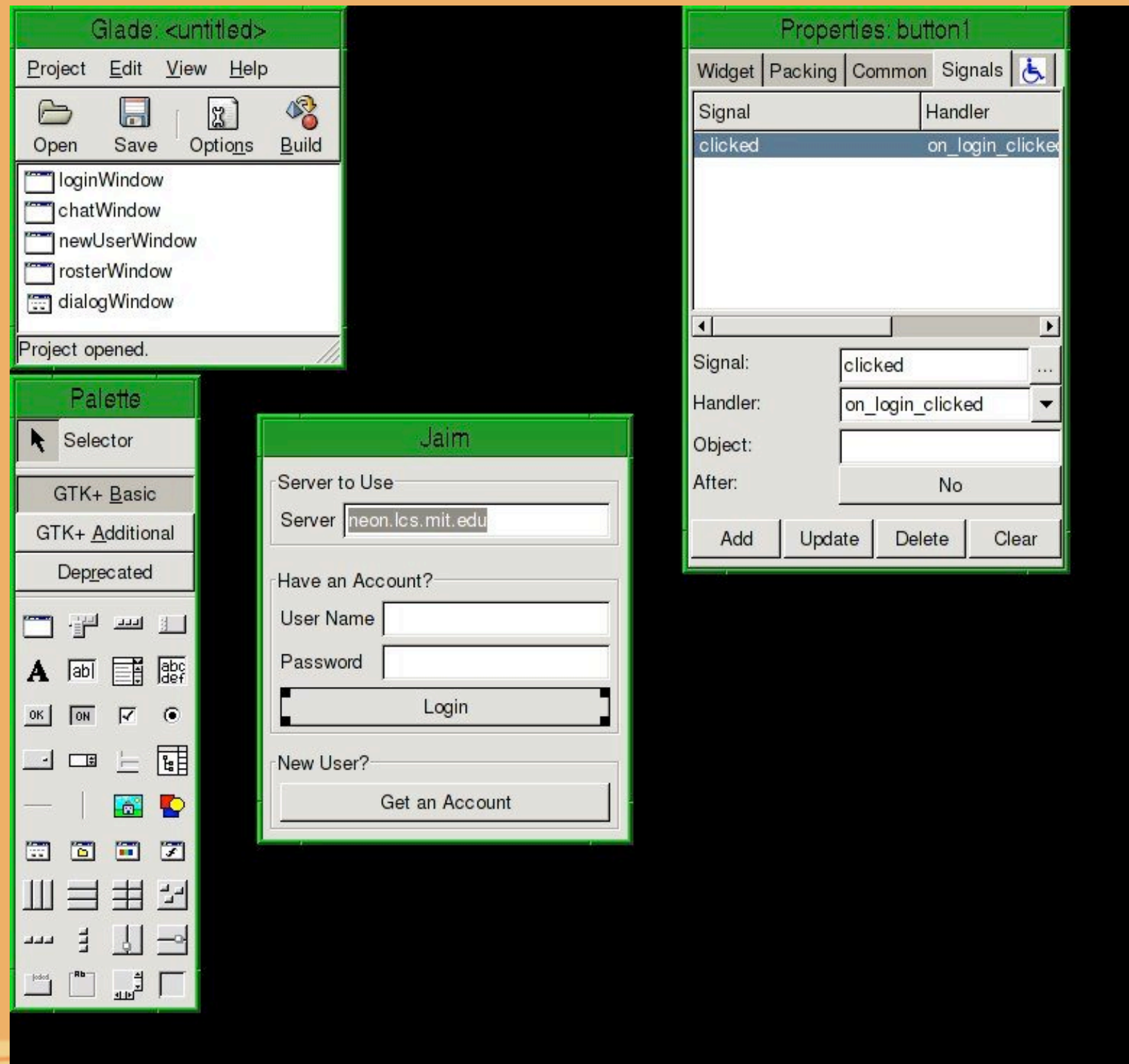
● **Glade** is a program designed to enable the quick building of graphical user interfaces for GTK+ and Gnome applications.

<http://glade.gnome.org/>

[http://www.kplug.org/glade_tutorial/glade2_tutorial/
glade2_introduction.html](http://www.kplug.org/glade_tutorial/glade2_tutorial/glade2_introduction.html)



Glade Screen Shot





GUI \Leftrightarrow App

- Application registers call-backs with GUI
- Signals or events invoke the call-backs
- Who ensures that the right connection?
 - It is all in the programmer's head
 - (generally not a good idea)
- How much does the GUI know about the semantics of the input?
 - Not very much
- Want abstraction and autogen or checker
 - still open problem





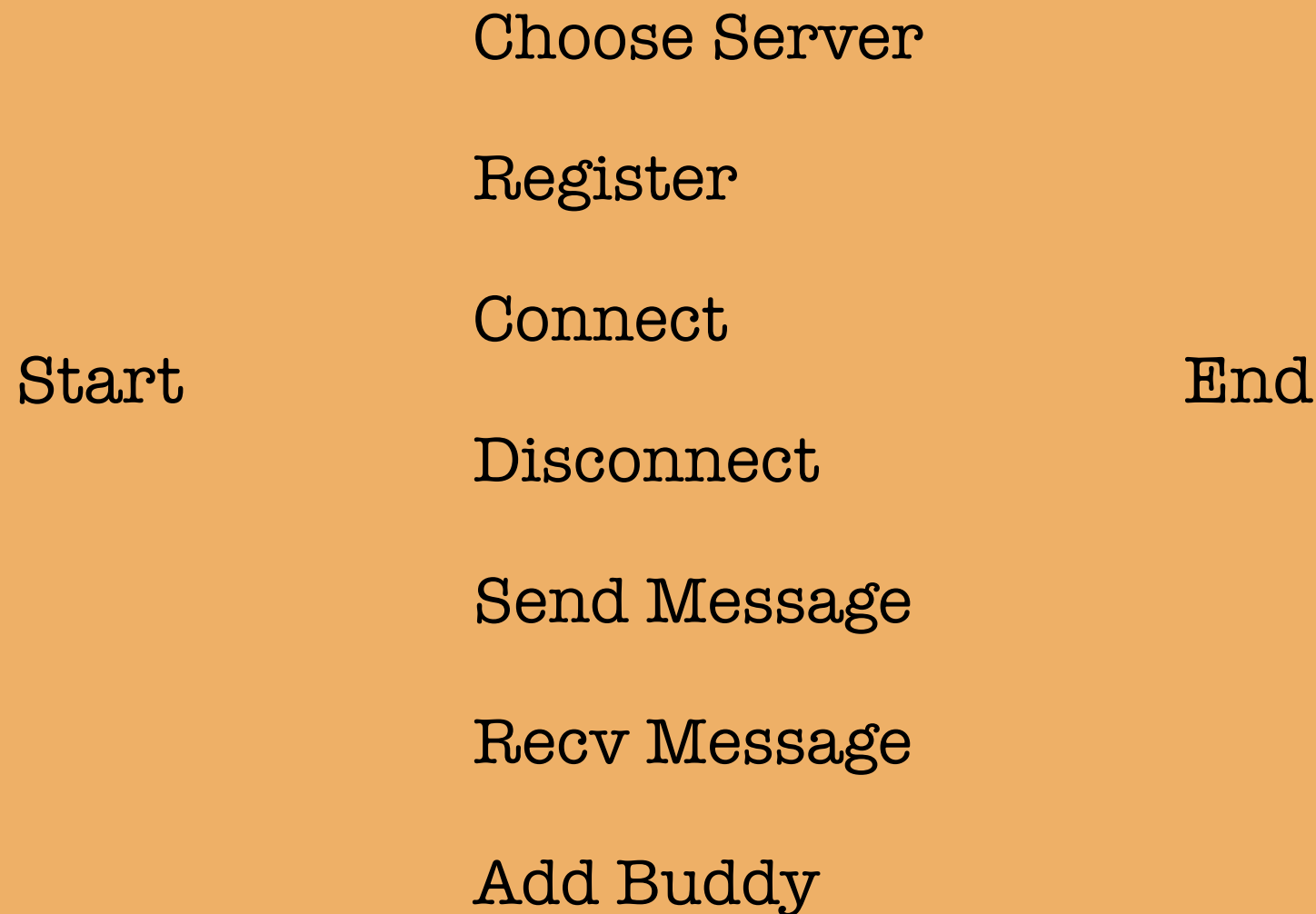
Instant Messaging

- A server mediates messages
- Clients registers and connect with server
- Clients “subscribe” to other clients
- Client sends message to server, server forwards to recipient
 - can only send messages to registered/published recipients
- Supported operations:
 - Register, Unregister, Logon, Logoff, Send/Rec, Add/Remove Buddy, etc.





IM State Transition Diagram





IM State Transition Diagram II

Choose Server → Register → Connect ▲ Disconnect →

Start

End

Send Message

Recv Message

Add Buddy





Better Specification of UI

- Want universal user interface description
- Application can interact with several UI's
 - keyboard, mouse, touch screen
 - speech, gesture
 - agent (a real PDA)
 - brainwaves, ESP
- Answer: still open research question





The challenge of handhelds

- Small screen; poor lighting, stylus
- Touch screen
 - on-screen keyboard
 - handwriting recognition
 - menu's
 - jog dial
- What is the problem?



Input Errors

- Touch screen
 - on-screen keyboard: hit adjacent letters
 - handwriting recognition: mis-recognition
 - menu's: choose adjacent item
 - jog dial: choose adjacent item
- Select wrong icon
 - Cannot see the writing (old folk's problem)
- Just do the wrong thing



What can be done?

- Add semantic knowledge
 - Part of UI specification
 - Limit acceptable input
- User interface can pass on several alternatives to application
 - Top “N” choices with associated probability
 - Application can handle probabilistic choice of input





Errors are device specific

- Adjacent letter mistake
 - what do you think “ues” means
 - does it matter if it is keyboard or xstroke?
- xstroke has reg-exp description of letters





Xstroke

```
# 1 2 3
# 4 5 6
# 7 8 9
```

```
# The extents of the grid will be automatically inferred based on the
# bounding box of the input stroke. This makes xstroke robust to many
# stroke distortions including translation and independent scaling
# along the X and Y axes.
```

```
#
```

```
# For example, an intuitive stroke for the letter L might be:
```

```
#
```

```
# Key L = 14789
```

```
# Key L = 147?89
```

What letter is this?

```
([12]*[45][78]I[12][45]+[78]?)?[78]*[4]*(1?[2][369]+I1[25][369]*)([369]+[25]+
8?[147]?[258]*[369]+I[25]*8?[147]+[258]+[369]*)([369]*[58][74]+I[369]+[58][74]*)
```





What do we need?

- “Learning” (AI techniques)
- Bayes Theorem
 - what is the probability that the user meant
 - X given this input (“yes” given “ues”)
 - Y given this input (“no” given “ues”)
 - Z given this input (“help” given “ues”)
 - can figure this out from the opposite way:
 - the probability that the user typed “ues” given that she wanted to type “yes”

