GUI Design

Brian H. Toby
NIST Center for Neutron Research

Outline

• Background
• Script languages for GUIs
• GUI design do’s & don’ts

Why use a GUI?

GUI = Graphical User Interface

• A well-designed GUI speeds learning
  – Opens software to occasional users & novices
• Scaleable: offers power tools to experts

Portable GUIs

Windows only?
Support for Linux & Mac offers wider range of users & growth into parallel processing

Portable GUI tools

• Compiled (usually C++) packages
  – FLTK (www.fltk.org)
  – wxWidgets [nee wxWindows] (www.wxwidgets.org)
• Virtual Machine
  – Java
• Script languages
  – Python + Tk, +wxWidgets, +GTK
    • GUI Builders: wiki.python.org/moin/GuiProgramming
  – Tcl/Tk (www.tcl.tk & comp.lang.tcl)
Pros & cons of scripting

Pros

• Easy to code
• Test small routines
• Extensible when speed is needed
• Highly portable
• Add code at run time

Cons

• Slower than compiled code
• Debugging can be non-trivial

IMHO 1: GUIs do not need tremendous speed

• GUIs interact with people, who cannot tell the difference between a 10 µsec vs a 50 millisecond screen paint

IMHO 2: Where possible don’t incorporate code into script language, use external programs

When more extensive computations are needed, one can pass information to an external program, run it & read back results

– More portable
– Easier to debug
– More than fast enough: overhead of write, fork & read is usually trivial

Example: Calling an external program

CMPR EditCell replaced SGI GL program
MANDEX: animate powder diffraction line positions

• 1st draft: run FORTRAN program each time slider is moved
  – Fast enough
• Final version: modify FORTRAN output for direct parsing by interpreter
  – Even faster!

Thoughts on GUI design

1st vs. 2nd gen. output

<table>
<thead>
<tr>
<th>ICD</th>
<th>H</th>
<th>K</th>
<th>L</th>
<th>MULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

set dgen1(k) {
  17.7129707 22.191597 28.5358257 29.7368317
  34.8244629 35.8678513 37.4165802 41.6833496
  42.5831184 45.2750664 47.2591591 48.9819603
}

set dgen1(h) {
  0
  1
  0
  0
  1
  1
  0
  1
  1
}

set dgen1(k) {
  0 0 1 1 1 0 1 1 0 0 2
}

set dgen1(l) {
  1 1 0 1 1 2 1 1 2 2 1
}
2nd Generation GUIs depend on a visual short-hand

- Analogy to physical things
  - Push Buttons/Toggle buttons
  - Notebook tabs
- Last operation in bottom corner
  - MS: Save/Cancel
  - Motif: Apply/Accept/Cancel
- GUIs often follow visual conventions

Other Design Goals

- Screen space is valuable - don’t waste it
- A little bit of color helps guide the eye
- Too much color is confusing
  - Keep contrast levels high (have pity on the color blind)
- Try to use color consistently

GUI design: Hall of Fame & Shame

CIFEDIT: not simple but easy

Another of my “greatest hits” EXPGUI
Multi-step processes are tough with GUIs.
I don’t know how to make this more intuitive

Even worse
Even worse

Conclusions

• Script languages are great for portable GUI design
• Intuitive GUIs take considerable thought
• Use conventional designs where possible
• Multi-step procedures are tough to make intuitive
  – Tutorials help
• Users really like GUIs