How to add (simple) interactivity to Madagascar: A proposal

Joe Dellinger, BP
What do I mean by “interactive”?

• If you want to do REAL interactive graphics, you probably need to create your masterpiece!

• I’m thinking of programs like “sfgrey” and “sfgraph”, not a velocity-analysis-picking tool.

• My goal is to enable some simple useful interactivity requiring minimal coding by the Madagascar user.

• Respect Madagascar’s core philosophies:
  – Simplicity (at most a few lines of code)
  – Modularity (pieces should work together)
  – Device independence (i.e. don’t edit a bitmap)
  – Reproduceability (i.e. not one-off works of art)

• Don’t break things!
Why this talk?

- I think this is the right way to attack the problem!
- Feedback / help requested!
The structure of a figure

What do we have here?
• Nested groups of objects
• Axes, labels, rasters

This figure was made by a script.
That script used parameters to put everything in the right places.

Optimizing script parameters is a pain.
What might you do to this figure?

Graphical things to do:
• Toggle elements on / off
• Change a label
• Resize some text
• Move a box
• Add annotation
What might you do to this figure?

Scientific things to do here:

• Change axis limits for a different view
• Move one of the zoom-in boxes
• Get coordinates of a point
• Read off a value
Overall philosophy

Parameter files

Script

Program1 par=....
Program2 par=....
Program3 par=....

“pen” filter

Vplot file
Overall philosophy

Parameter files

User input

“pen” filter

Script

Vplot file

Program1 par=....
Program2 par=....
Program3 par=....
Current Vplot Methodology

- Elements of figures are in nested groups
- Each group can have a name
- We don’t do anything useful with groups now

- Pen filter option: interact=file_name
- Click and the vplot coordinates of that point (in inches) are written to file_name
Proposed New Methodology

- Elements of figures are in nested groups
- Each group can have a name

- That “name” can be a long text string.
- We can use that text string to leave instructions in the vplot file telling the “pen” program what to do.
- Will look like a command line.

- Could write out a parameter file. Where?
  - interact.Xaxis=axis1.par
FORMAT controls the output. Interpreted sequences are:

- `%%` a literal `%`
- `%a` locale’s abbreviated weekday name (Sun..Sat)
- `%A` locale’s full weekday name, variable length (Sunday..Saturday)
- `%b` locale’s abbreviated month name (Jan..Dec)
- `%B` locale’s full month name, variable length (January..December)
- `%c` locale’s date and time (Sat Nov 04 12:02:33 EST 1989)
- `%C` century (year divided by 100 and truncated to an integer) [00-99]
- `%d` day of month (01..31)
- `%D` date (mm/dd/yy)
- `%e` day of month, blank padded ( 1..31)
- `%f` same as `%Y-%m-%d`
- `%g` the 2-digit year corresponding to the %V week number
- `%G` the 4-digit year corresponding to the %V week number
- `%h` same as `%b`
- `%H` hour (00..23)
- `%I` hour (01..12)
- `%j` day of year (001..366)
- `%k` hour ( 0..23)
- `%l` hour ( 1..12)
- `%m` month (01..12)
- `%M` minute (00..59)
- `%n` a newline
- `%N` nanoseconds (000000000..999999999)
- `%r` time, 12-hour (hh:mm:ss [AP]M)
- `%R` time, 24-hour (hh:mm)
- `%s` seconds since ‘00:00:00 1970-01-01 UTC’ (a GNU extension)
- `%S` second (00..60); the 60 is necessary to accommodate a leap second
- `%t` a horizontal tab
- `%T` time, 24-hour (hh:mm:ss)
- `%u` day of week (1..7); 1 represents Monday
- `%w` day of week (0..6); 0 represents Sunday
- `%w` week number of year with Sunday as first day of week (00..53)
- `%W` week number of year with Monday as first day of week (01..53)
- `%y` last two digits of year (00..99)
- `%Y` year (1970..)

Inspiration:

date +’%A %B %e %Y’

Produces:

Thursday June 16 2011
How to do it?

• Each group has a name
  name="Xaxis"

• Each group specifies an “active region”
  – Usually will just be a box
  – May be a circle around a point
  – Or a user-defined boundary
    boundingbox=xmin,xmax,ymin,ymax

• Each group has a precedence
  – On top (default)
  – Or on bottom

• And an opacity
What might you do to this figure?

Possible actions:
- Request user input
- Print value to screen
  - default area
  - area just for this group
- Print to file
- Change precedence
- Change opacity
- Delete this group
- Rewind and replot
- Exit
What to print?

• Actions:
  – Hover
  – Click (1, 2, 3?)
  – Click and drag (1, 2, 3?)
  – Print prompt and accept text input

• Available output values in “printf”:
  – Vplot X, Vplot Y
  – Text string associated with color table value associated with raster position
  – Which button was clicked
  – Prompted text input
  – Value projected along an axis
  – Specify arbitrary formula?
What needs to be done?

- Begin group, end group commands already exist
  - Need to gracefully handle long group “names” (sfplas, sfpldb)
- New command to associate a text string with a raster color table value

- Example “wrapper” script
- Add interactivity to existing graphics programs as examples
- Start with “show value on hover”

- Only hard work is in dovplot.c
What needs to be done?

• Need CONVENTIONS!

• Like, “right click to find out what’s possible here”? 
Example: “add_label.vpl”

- Contains group that “covers entire screen” (default)
- Opaque group
- Contains commands for click and drag
- Prompts for text box on click and drag

- Simple script appends “label.vpl” onto existing file to add annotation:
  - Pen erase=once interact.label=parfile  plot.vpl label.vpl
- Writes out parfile for desired label
What could we do with this?
Thanks!

- I think that’s probably enough for a morning talk…
- Thanks to everyone helping keep the old SEPlib collaborative spirit alive!