Palm OS Programming - Gotchyas

This is the text, extracted from a presentation for Palm OS Programming Advanced UI elements. The gotchyas and pitfalls were skipped because I already had that in a separate page. I don't use Adobe and I wanted a textual reference instead of printing out slides, so I did this. View the original information at PalmSource

---

Anatomy of the Stock Event Loop:

```
EventType event;
do
{
    EvtGetEvent(&event, evtWaitForever);
    if (! SysHandleEvent(&event))
        if (! MenuHandleEvent(0, &event, &error))
            if (! AppHandleEvent(&event))
                FrmDispatchEvent(&event);
} while (event.eType != appStopEvent);
```

- FrmDispatchEvent passes the event FIRST to the handler that was registered (usually by AppHandleEvent)
- If the event wasn't handled (i.e. the handler returns false) it flows to FrmHandleEvent, where it receives default handling
    - Keystrokes flow into fields
    - FrmUpdateEvent calls FrmDrawForm()

Opaque UI Structures

- Myth:
    - UI structures are documented so you can edit their contents.
- Reality:
    - UI structures are documented to help debugging.

Usable and Visible

- Myth:
    - Applications can control which form objects draw by setting the "visible" bit.
- Reality:
    - The "visible" bit is actually state information, not something you should be setting.
    - Use FrmShowObject() and FrmHideObject()

Clipboard

- Myth:
    - The clipboard is just for boring text.
- Reality:
    - You can put bitmaps on the clipboard, too! ClipboardAddItem(clipboardBitmap, bitmapP, MemPtrSize(bitmapP));
    - Text and bitmap clipboards are separate; one doesn't overwrite the other

Custom Fonts

- Myth:
    - Custom fonts aren.t supported.

- Reality:
  - Custom fonts ARE supported. We don.t make it easy (yet)..

Creating a Custom Font

- Fonts are NFNT resources (well, almost)
  - Slight differences in header
  - Glyphs contain space
- No support in Constructor for creating fonts
- But there are various third-party tools for creating and importing fonts
  - xFont
  - PilRC
  - ResEdit (Macintosh)

Adding a Font to your Project (PC)

- Use xFont (freeware) to make a font in PilRC format. Call it something like "myFont.pfn"
  - Or create font directly in PilRC format - See PilRC Manual for details
- Install PilRC CodeWarrior plugin
  http://www.calliopeinc.com/pilrcplugin.html
- Add a .rcp file to project with this line in it:
  - Font 'FONT' FONTID 128 "myFont.pfn"
- Compile and get a .r file containing a "NFNT" resource
  - data 'NFNT' (20052) means Resource ID is 20052

Installing a Custom Font

```
MemHandle fontH;
FontType* fontP;
fontH = DmGetResource(fontResource, 25002);
fontP = (FontType*) MemHandleLock(fontH);
FntDefineFont(fntAppFontCustomBase, fontP);
MemHandleUnlock(fontH);
DmReleaseResource(fontH);
```

- fntAppFontCustomBase is the first ID available
  - add an offset when defining multiple fonts

Using a Custom Font

- WinDrawChars will now use this font
  - if you call FntSetFont(fntAppFontCustomBase)
- LstGlueSetFont, FrmGlueSetLabelFont, CtlGlueSetFont, TblSetItemFont can then take fntAppFontCustomBase to set the list, form, control or table font to the new custom font

High Resolution Font Families

```
 LowRes
NFNT 2000  ---->      Constructor
                          or             ---->   Family
  HiRes    ---->  tfnf resource defines         NFNT 4000
NFNT 3000          font family members
```

- Use only on Palm OS 5 and later
  - Not backwards compatible (check OS version!)
- DmGetResource('nfnt', myFontFamilyID);
- From there use same as "old style" custom font

Tables

- Myth:
  - Tables are Complicated
- Reality:
  - Tables are Very Complicated

Do You Really Need a Table?

- Tables are ideal when:
  - You need UI widgets embedded in a table
    - fields, checkboxes, or anything that accepts input
- Consider alternatives when:
  - You simply want to display data in a multi-column format
  - You require scrolling
    - Tables have no inherent scrolling functionality!

Alternatives to Tables

- Lists
  - Can have multiple columns when drawn by callback
  - Great when rows can select together
- Fields + Scrollbar
  - Great for text-only data
- Gadgets
  - Most flexible, most work
  - Get a rectangle that receives hits
  - Implement only the functionality you need

Attention Manager

- Myth:
  - The Attention Manager is a 4.0 replacement for the Alarm Manager.
- Reality:
  - The Attention Manager is a central UI repository for attention getting messages from all applications.

Step 1: Call the Attention Manager

- Often in response to an Alarm or Notification

```
AttnGetAttention(cardNo, dbID, userData, NULL,
                 kAttlLevelSubtle, kAttnFlagsUseUserSettings, 5, 2);

// Handle the launch code sysAppLaunchCmdAttention

case sysAppLaunchCmdAttention:
    HandleAttention((AttnLaunchCodeArgsType*)cmdPBP);
```

Step 2: Draw the Attention UI

```
switch (cmdPBP->command)
    case kAttnCommandDrawList:
        // Draw the item in the list
    case kAttnCommandDrawDetail:
        // Draw the detail screen

/// Code to help you out
x = cmdPBP->commandArgsP->drawList.bounds.topLeft.x;
```

```
y = cmdPBP->commandArgsP->drawList.bounds.topLeft.y;
WinDrawBitmap(iconP, x, y);
x += kAttnListTextOffset;
WinDrawChars(theStr, StrLen(theStr), x, y);
```

Step 3: Extras

- case kAttnCommandPlaySound
    - Chance to play a custom sound
    - Sent if requested in AttnGetAttention() call
- case kAttnCommandCustomEffect
    - Chance to do any other custom effect

Step 4: Handle User's Choice

```
case kAttnCommandGotIt:
    // Selected the OK button

case kAttnCommandSnooze:
    // Selected the Snooze button

case kAttnCommandGoThere:
    // Selected the Go To button
    AttnForgetIt(cardNo, dbID, cmdPBP->userData);
    buf = (UInt32*) MemPtrNew(sizeof(UInt32));
    MemPtrSetOwner(buf, 0);
    *buf = paramP->userData;

    SysUIAppSwitch(cardNo, dbID,
                   sysAppLaunchCmdCustomBase, buf);
    // Handle sysAppLaunchCmdCustomBase in PalmMain()
```

Step 5: Update the Attention Manager

- AttnForgetIt -- when something in the attention manager is no longer relevant
- AttnIterate -- Iterates through everything that belongs to your app and gives your callback a chance to handle each one
    - Update all attentions
    - Erase all attentions
- AttnUpdate -- Update specific attention
- AttnGetCounts -- Use to find out how many attentions you have pending

Tyler Akins <fidian@rumkin.com>