An Advanced Graphics Platform for Embedded Linux

Robi Karp, Fluffy Spider Technologies
“Almost every single consumer facing electronic device requires a graphical user interface” - Me.
About FST

- Founded 1995
- Embedded systems specialist since 1995
- Linux 1995
- Based in Sydney, Australia
- Product license sales
  - FancyPants to VeriFone International
- Custom development
  - Embedded Linux / GUI
  - Toshiba, Vodafone, Taxitronic, HPM, Optus, Telstra
- Support
  - Support Toshiba for 7 years for each product
The history of the user interface on a typical consumer electronic device:
Market Trends

Richer user interfaces
- Less constraints
- More “eye-candy”
- More movement

Further use of video
- Video searching
- Video manipulation
- Object recognition (for still images too)
Market Trends Example: Kiosks
Market Trends

❖ Fashion
❖ "Style"
FancyPants Beginnings
E17: [http://www.enlightenment.org/](http://www.enlightenment.org/)

- Open Source project developing the “next generation” desktop for Linux.

- Large project with many libraries.

- Main libraries used by FancyPants are:
  - Evas – display canvas
  - Ecore – core event loop, ipc, signals, etc.
Evas

- Display canvas
- Non immediate mode
- Compositing with true alpha blending
- Tiling
- Abstracted backends
- Small and fast, suited to embedded

To (sort of) quote Han Solo:
“*We've made a lot of special modifications ourselves.*”.

FST works together with the E17 development community.
The Enlightenment desktop project has gone through its own evolution:
What Is FancyPants?

▶ GUI Platform

▶ Multi Media
What Else Is FancyPants?

► Designed for embedded
Portability + Ease of Development

Display System Abstraction:
- Linux / X11
- Linux / Framebuffer
- Linux / DirectFB
- Linux / Qtopia

OS Abstraction:
- Symbian / UIQ
- Win32 / PocketPC / WinCE (v4.x)
- L4

Important for development: Portability + Ease of Development
### Technical Details: Size

<table>
<thead>
<tr>
<th>Module</th>
<th>Install Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. FancyPants Graphics</td>
<td>650 KB</td>
</tr>
<tr>
<td>2. FancyPants Media</td>
<td>500 KB - 1.5 MB  [*]</td>
</tr>
<tr>
<td>3. FancyPants Canvas Server</td>
<td>40 - 50 KB</td>
</tr>
</tbody>
</table>

**TOTAL:**

650 KB - 2.2 MB

[*] Depending on codecs
Canvas Server

- Very very small: 40-50KB install
- Minimal – in line with FancyPants philosophy
- Across multiple virtualised operating systems
Canvas Server: Multiple Operating Systems

- FancyPants App
- Linux Virtualised
- Other OS (Linux, eCos)
- Host OS - L4
<table>
<thead>
<tr>
<th>API</th>
</tr>
</thead>
<tbody>
<tr>
<td>• C</td>
</tr>
<tr>
<td>• Bindings</td>
</tr>
<tr>
<td>• Minimal design philosophy</td>
</tr>
<tr>
<td>• No wrapping “for completeness”</td>
</tr>
<tr>
<td>• About 20 widgets (included in 650KB)</td>
</tr>
<tr>
<td>• Simple way to create new widgets; Smart Widgets</td>
</tr>
<tr>
<td>• Exposed Evas API</td>
</tr>
<tr>
<td>• OO interface; Use Evas_Object</td>
</tr>
<tr>
<td>• Layers</td>
</tr>
<tr>
<td>• Grouped Layers via Smart Widgets</td>
</tr>
</tbody>
</table>
Example Media Player Code:

```c
#include <stdio.h>
#include <Ecore_Evas.h>
#include <fp.h>
#include <media/fp_media.h>

int fp_main(int argc, char **argv)
{
    struct fp *fp;
    Evas_Object *bg;
    Evas_Object *media_obj;

    fp = fp_init(&argc, argv);
    fp_window_init(fp, "");

    media_obj = fp_media_add(fp);
    fp_media_file_set(media_obj, argv[1]);
    evas_object_show(media_obj);
    fp_media_audio_enabled_set(media_obj, 1);
    fp_media_state_set(media_obj, FP_PLAY);
    fp_begin(fp);
}
```

Same program for audio and for video

Simple:

1) Add generic object
2) Press "PLAY"
Example Media Player Makefile

CFLAGS += `pkg-config --cflags fancypantsmedia`
LDFLAGS += `pkg-config --libs fancypantsmedia`

demo1: demo1.c
Example: Resizing Media

```c
media_obj = fp_media_add(fp);
fp_media_file_set(media_obj, argv[1]);
evas_object_resize(media_obj, 400, 800);
evas_object_show(media_obj);
fp_media_audio_enabled_set(media_obj, 1);
fp_media_state_set(media_obj, FP_PLAY);
```
Example: Resizing Media

```c
media_obj = fp_media_add(fp);
fp_media_file_set(media_obj, argv[1]);
fp_media_video_keep_aspect_set(media_obj, 1);
evas_object_resize(media_obj, 400, 800);
evas_object_show(media_obj);
fp_media_audio_enabled_set(media_obj, 1);
fp_media_state_set(media_obj, FP_PLAY);
```