# GFXprim library introduction

## "minimalistic graphics and widget library"

Cyril Hrubis

6. February 2021

GFXprim

# About GFXprim

- Minimalistic software 2d graphics
- Includes a widget library
- Written for Linux

  Quite a lot of the code should be OS independed
- LGPLv2.0+

GFXprim

# What makes it unique?

- Optimized for speed and low memory consumption

  Apps can run fine on 300Mhz PXA270 with 32MB of free RAM

  Stripped libraries around 1MB in size

- The dependencies are fairly minimal

  most can be disabled upon compilation

- Modular and flexible

  Apps can run, without any changes, under X11, framebuffer, etc.

GFXprim

# Why is GFXprim unique?

- C code for different pixel types and sizes is generated

  Uses a python based template engine

- Sane widget API that's not riddled with abstraction

GFXprim

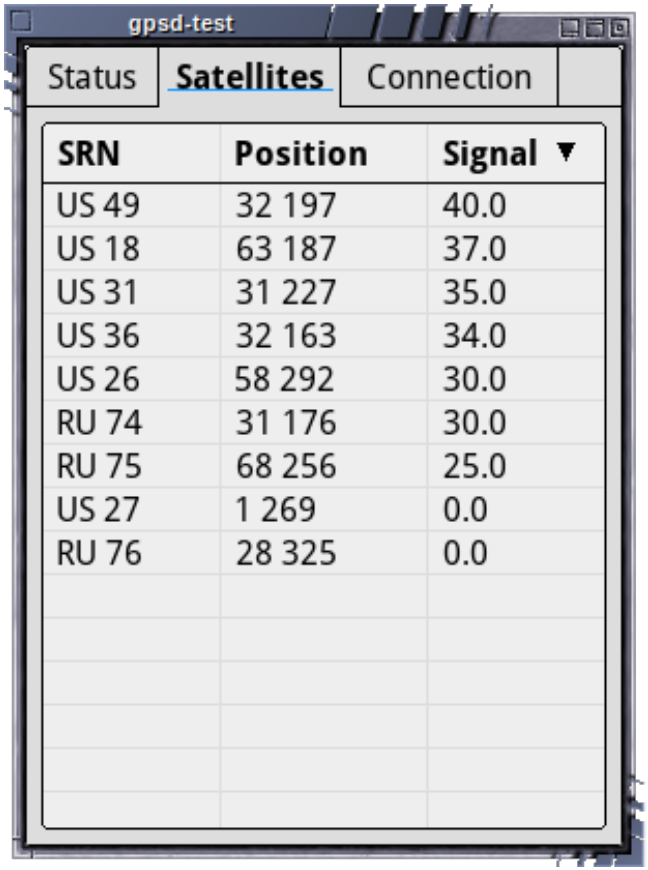
# When and how it all began?

- Somewhere around 2010 I've played with sharp zaurus PDAs
- I had Debian and upstream kernel running
- X11 was not usable, etc.
- I wanted GUI something like PalmOS had
- It's my "toy" project for a bit more than 10 years now

GFXprim

# When and how it all began?

Fast forward to 2020 it's finally there.





GFXprim

# What are the usecases?

The initial end goal was fast GUI on embedded hardware however these days GFXprim can be used for:

- Drawing basic graphic primitives, e.g. line, circle, text, etc.
  support anti-aliasing is on TODO though
- Conversion between different image formats
  jpg, png, webp, tiff, gif, bmp, pnm, jp2, pcx, psp, psd, ico
- Image processing library
  e.g. point filters, convolutions, dithering, resamplings...

GFXprim

# What are the usecases?

The initial end goal was fast GUI on embedded hardware however these days GFXprim can be used for:

- Graphics library that can draw on and get input from X11, framebuffer, aalib, SDL, Linux input
- Widget library in C that has (hopefully) sane API

GFXprim

# Widgets

The widget library implements:

- basic widgets

  buttons, radiobuttons, sliders, checkboxes, text input, tables, etc.
- dynamic layouts
- layouts from JSON
- dialog support

GFXprim

# Widgets

```
{
 "version": 1, "cols": 2, "rows": 2,
 "widgets": [
  ...
  {
   "rows": 2,
   "widgets": [
    {
     "type": "tbox", "text": "admin",
     "uid": "uname", "on_event": "login_callback"
    },
    {
     "type": "tbox", "hidden": true,
     "uid": "pass", "on_event": "login_callback"
    }
   ]
  },
  {
   "type": "button", "label": "Ok",
   "on_event": "login_callback"
  }
 ]
}
```

GFXprim

# Widgets

```
static gp_widget *pass, *uname;

int login_callback(gp_widget_event *ev)
{
    if (ev->type != GP_WIDGET_EVENT_WIDGET)
        return 0;

    if (uname)
        printf("Username: '%s'\n", uname->tbox->buf);

    if (pass)
        printf("Password: '%s'\n", pass->tbox->buf);

    return 0;
}
```

GFXprim

# Widgets

```
int main(int argc, char *argv[])
{
    void *uids;

    gp_widget *layout = gp_app_layout_load("login-example", &uids);
    if (!layout)
       return 1;

    pass = gp_widget_by_uid(uids, "pass", GP_WIDGET_TBOX);
    uname = gp_widget_by_uid(uids, "uname", GP_WIDGET_TBOX);

    gp_widgets_main_loop(layout, "Login!", NULL, argc, argv);

    return 0;
}
```

GFXprim

# Links

- Home Pages:
  `http://gfxprim.ucw.cz`
- GIT repository:
  `https://github.com/gfxprim/gfxprim`
- Mailing list:
  `https://www.ucw.cz/mailman/listinfo/gfxprim`
- Packages:
  `http://gfxprim.ucw.cz/packages.html`

GFXprim