Rendering SVG graphics with libSDL

Presented by:
Bill Kendrick
<bill@newbreedsoftware.com>
Lead Developer, Tux Paint
Tux4Kids
http://tux4kids.alioth.debian.org/

SVGOOpen Conference
October 3, 2009
What is libSDL?

Simple DirectMedia Layer library
GNU LGPL license
Multi-platform:
• Linux, *BSD, Solaris, IRIX, QNX
• Windows, Windows CE
• Mac OS X, Mac OS
• Amiga, iPhone, Dreamcast, Atari ST, AIX, RISC OS, SymbianOS, OS/2
What is libSDL? (cont'd)

- Written in C
- Works with C++
- Bindings to: Ada, C#, D, Eiffel, Erlang, Haskell, Java, Lisp, Lua, ML, Objective C, Pascal, Perl, PHP, Python, Ruby, Smalltalk, Tcl and more...
- Used in open source and commercial games
What is libSDL (cont'd)
Images in libSDL

• SDL has built-in BMP loading capabilities
• ”SDL_image” add-on library allows loading many more bitmap formats:
  – GIF, JPEG, LBM, PCX, PNG, PNM, TGA, TIFF, XCF, XPM, XV
• Notice no SVG...
Why I needed SVGs

Tux Paint! (My pet project)
Why I needed SVGs (cont'd)

Tux Paint!

- Open source drawing program for kids
- Includes "Stamps" feature to place pre-drawn art and photographs into pictures

"Stop" by Jarno K., Finland, 2007
How Tux Paint got SVG love:

I had a lot of time commuting on Amtrak...

Photo: snty-tact on Wikipedia; GFDL, cc-by-sa-2.5
A quick dance with libraries...

- **Step 1:** `svg`, `cairo` and `svg-cairo` libraries
  - Available on the version of Debian GNU/Linux I was running at the time
  - Was deprecated by the time that Debian version stabilized
- **Step 2:** Re-write using `rsvg-2` and `cairo`
- **Happy inconvenience:** Older, less-perfect SVG libraries allow us to support SVG on older platforms (e.g., RedHat Linux 9)
How it's done (the modern way)

- **Initialize:**
  ```c
  rsvg_init();
  ```

- **Open the SVG image file:**
  ```c
  rsvg_handle = rsvg_handle_new_from_file(file, &err);
  ```

- **Acquire its dimensions:**
  ```c
  rsvg_handle_get_dimensions(rsvg_handle, &dimensions);
  ```

- **Determine its pixel dimensions:**
  ```c
  rwidth = dimensions.width;
  rheight = dimensions.height;
  ```
How it's done (cont'd)

- Decide how to scale it to fit in Tux Paint's canvas:
  \[
  \text{scale} = \text{pick\_best\_scape}(\text{rwidth}, \text{rheight}, \text{r\_canvas.w}, \text{r\_canvas.h});
  \]
  /* An internal Tux Paint function, also used w/ PNG stamps */

- Apply the scale:
  \[
  \text{width} = ((\text{float}) \text{rwidth} \times \text{scale});
  \text{height} = ((\text{float}) \text{rheight} \times \text{scale});
  \]

- Create a buffer into which we render the SVG drawing:
  \[
  \text{stride} = \text{width} \times 4; \quad /* 4 \text{ bytes/pixel (32bpp RGBA)} */
  \text{image} = \text{calloc}(\text{stride} \times \text{height}, 1);
  \]

- Use it as a Cairo surface
  \[
  \text{cairo\_surf} = \text{cairo\_image\_surface\_create\_for\_data}\left(\text{image, CAIRO\_FORMAT\_ARGB32}, \text{width, height, stride}\right);
  \]
How it's done (cont'd)

- Create a new Cairo object:
  ```c
  cr = cairo_create(cairo_surf);
  ```

- Give it the scale value (so it fits within our scaled buffer):
  ```c
  cairo_scale(cr, scale, scale);
  ```

- Tell RSVG to render the SVG into the Cairo buffer:
  ```c
  rsvg_handle_render_cairo(rsvg_handle, cr);
  ```

- All done rendering!
  ```c
  cairo_surface_finish(cairo_surf);
  ```
How it's done (cont'd)

- Create an SDL surface to pass back to Tux Paint:
  rmask = 0x00ff0000;
gmask = 0x0000ff00;
bmask = 0x000000ff;
amask = 0xff000000;

  /* (Notice it matches CAIRO_FORMAT_ARGB32) */
  sdl_surface = SDL_CreateRGBSurfaceFrom(
    (void*) image,
    width, height,
    32 /* 4 bytes/pixel = 32bpp */,
    stride,
    rmask, gmask, bmask, amask);

- Of course, there's also the error checking and clean-up...
The Result!

Your image has been saved!
The Result!
SVG adoption in Tux Paint Stamps (versus PNG)
Links for Info & Downloads:

- Simple DirectMedia Layer library (libSDL)
  http://www.libsdl.org/
- GNOME SVG library (librsvg)
  http://librsvg.sourceforge.net/
- Cairo library
  http://www.cairographics.org/
- Tux Paint
  http://www.tuxpaint.org/
  (find these slides under “Events”)