

# The Magic Behind Tux Paint

- Bill Kendrick, Lead Developer

All of the important things Tux Paint does behind the scenes, and transparent to the users.

Sacramento Linux Users Group

June 11, 2009

# Background: What's Tux Paint?

- Drawing program for children
- Open Source (GPL)
- Written primarily in C
- Targets the Simple DirectMedia Layer API
- Ports kept up-to-date for Windows (back to Win95), Mac OS X, BeOS and Linux
- Started in June 2002
- Part of “Tux4Kids”



# Background: What does it do?

- Drawing (brushes, colors)
- Lines & shapes
- Text
- “Rubber Stamps” (clipart/photos)
- “Magic” tools (variety)
- Save/Load/Print/Un-/Redo
- Easy to use
- Highly configurable



# What's the Magic?

- **Magic tools (duh!)**  
lots of visual effects  
plug-in API system
- **Text tool**  
Input methods  
bidirectional
- **Stamps**  
Scaled PNGs  
SVGs  
Tinting
- **UI text**  
Localized (80+ languages)  
Nicely rendered

# ***Note on presentation slides***



- I'll be interrupting these slides to dig into the source code in an editor and talk about them
- If you're not at this SacLUG meeting, sorry you missed it!

# Notable Magic Effects

- Bricks & Rails
- Calligraphy
- Flower & Tornado
- Real Rainbow
- Fisheye & Ripples
- Blur & Smudge



# Magic API

- Callback-driven API based on events: click, drag, release, color / tool change
- Write code in C, build as a shared object (e.g., “.so” on Linux).
- Use 'tp-magic-config' script to compile/install
- See a full presentation at:  
<http://tuxpaint.org/presentations/tuxpaint-magic-api.pdf>

# Text Tool – Input Methods

- Rolled out own
- Japanese
- Chinese (Traditional)
- Korean
- Thai

ひらがな



# Text Tool - Bidirectional

- Using FriBidi...
- Grab Unicode characters from libSDL
- Pass them along to Input Method handler
- Construct string of `wchar_t`[]
- Cast `wchar_t`'s to `FriBidiChar`'s
- Send it to `fribidi_log2vis()`
- Cast `FriBidiChar`'s back
- Display!

עברית English العربي

# Stamps – Scaled PNGs

- Bilinear scaling (optional)
- Sample 4 pixels around the 'in-between' spot (subpixel)
- Blend them, adjusting for how much they contribute (i.e., how close they are to the subpixel)
- Based on:  
<http://www.codeproject.com/cs/media/imageprocessing4.asp>
- See also:  
[http://en.wikipedia.org/wiki/Bilinear\\_filtering](http://en.wikipedia.org/wiki/Bilinear_filtering)

# Stamps - SVGs

- Old way: libcairo1, libsvg, libsvg-cairo (based on example SDL code in Cairo project, from 2004)
- New way: libcairo2, librsvg, librsvg-cairo
- Ask libraries to render the SVG into a memory buffer
- Create a new SDL surface from that buffer
- Let them scale cake!



# Stamp Tinting

- ... stuff I didn't write  
(Thanks, Albert!)
- We can look at it, at least.

# UI Text Localization

- gettext and .po files
- Python tools to manage Stamp localization (.po↔.txt)
- Bash scripts and PHP to maintain translation status on website  
<http://www.tuxpaint.org/help/po/>
- Some strings used for 'scoring' fonts for the Text tool (“can this font show characters important for the current locale?”)

# UI Text Rendering

- SDL\_ttf was the old way
- SDL\_Pango is the new way
- Wrapped SDL\_ttf “TTF\_Font\*” and SDL\_Pango “SDLPango\_Context\*” in a structure, “TuxPaint\_Font”.
- Wrote lots of #ifdef'd management code, then was mostly a matter of search and replace:  
*s/TTF\_Font/TuxPaint\_Font/g*

# Get Tux Paint

- Website  
<http://www.tuxpaint.org/>
- SourceForge.net  
<http://www.sf.net/projects/tuxpaint/>
- Download builds / Download/find packages  
<http://www.tuxpaint.org/download/>
- Download source from CVS repository  
<http://tuxpaint.org/download/source/cvs/>
- Mailing lists, IRC, etc., too!

# Thanks & happy ~~painting~~ coding

- Bill Kendrick  
[bill@newbreedsoftware.com](mailto:bill@newbreedsoftware.com)
- You can also find me at:  
the Linux Users' Group of Davis  
<http://www.lugod.org/>
- Day job:  
CTO & Lead Dev at:  
Smashwords, Inc.  
<http://www.smashwords.com/>

