

Tux Paint, Technically

Bill Kendrick

Peninsula Linux Users Group
December 8, 2005

Linux Users' Group of Davis
December 19, 2005

bill@newbreedsoftware.com

<http://www.newbreedsoftware.com/tuxpaint/>

Tux Paint Overview

- Drawing/painting application
- Audience: children ages 3 and older
- Open Source
- Runs on: Linux, Windows, Mac OS X, BeOS, ...
- Written in C
- Uses Simple DirectMedia Layer (SDL), libPNG, gettext, SDL_image, SDL_ttf, and SDL_mixer

But What *Is* Tux Paint?



- Drawing program
- Fun sound effects
- Pleasant & easy UI
- Cartoon mascot
- Educational
- Extensible / customizable

The Interface

- Toolbox, selector & color palette:
 - PNG bitmap backgrounds
 - PNG bitmap icons (with alpha)
 - TrueType font labels
 - Labels translated via gettext
 - All hand-coded and SDL event loop-driven
 - Example: `SDL_MOUSEMOTION` events trigger cursor context sensitivity (e.g., “hand” shape over buttons, crosshair shape over canvas)

The Interface (cont'd)

- Pop-up dialogs
 - Simple C “for”-loop used for animation
 - Separate event-handling function
 - Returns TRUE or FALSE, depending on what's clicked
 - Generic function – accepts various arguments for “Yes” / “No” labels, icons to display, etc.
 - Wrapper functions used to keep most function calls simple (e.g., “do_prompt()” calls “do_prompt_XYZ()” with dummy arguments)

The Interface (cont'd)

- Text entry tool
 - Uses `SDL_Timer()` callback to blink insertion cursor
 - *Needs help!*
 - Doesn't support non-English characters very well, yet!
 - Might need some hacking on (or around) SDL itself
 - Doesn't do word-wrapping
 - Unnecessary blits made when adding/deleting text
 - Cannot use arrow keys to move cursor
 - Yep, you do it all yourself when you use SDL! ☺

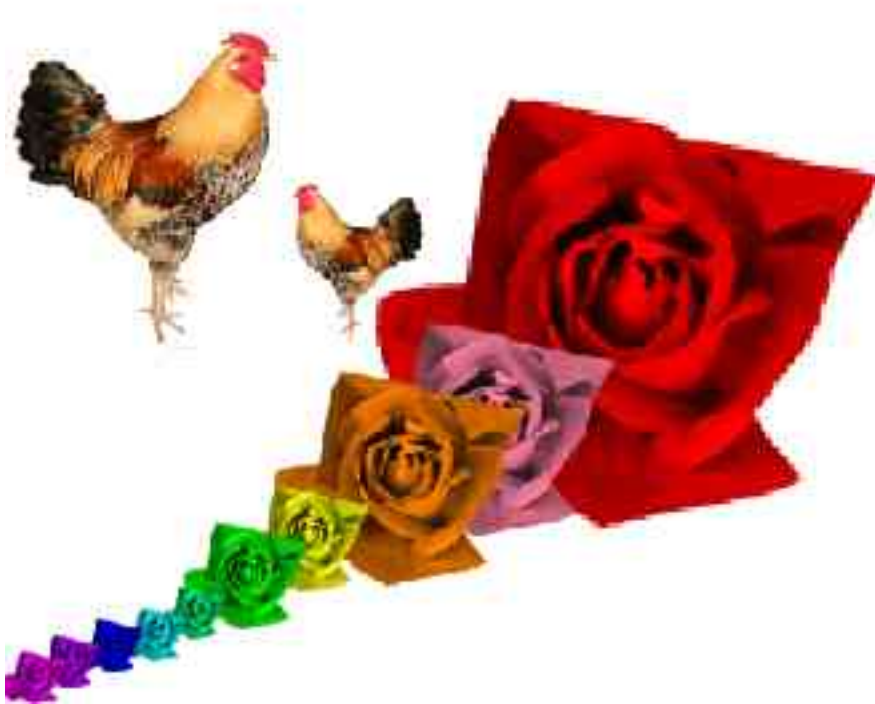
The Interface (cont'd)

- Word-wrapped text – blit one word at a time:

```
cur_x = x;
len = 0;
buf[0] = '\0';

for (i = 0; i <= strlen(str); i++)
{
    buf[len++] = str[i];
    if (str[i] == ' ' || str[i] == '\0')
    {
        /* ... use SDL_ttf to render word, then blit & free the surface ... */
        if (cur_x + surf->w > screen->w)
        {
            cur_x = x;                /* Back to left margin */
            cur_y = cur_y + surf->h;   /* Down one line */
        }
        cur_x = cur_x + surf->w;       /* Move cursor to end of the word */
        len = 0;                       /* Start a new word */
    }
}
```

Stamps



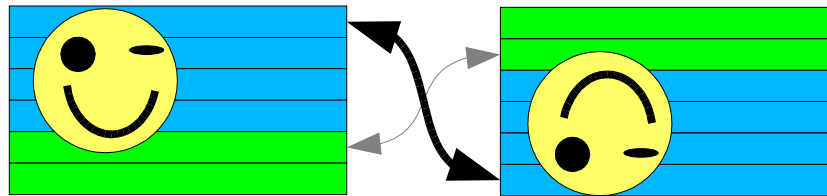
- PNGs with alpha
- WAV audio clips
- TXT descriptions
 - As of 0.9.15, translations stored in gettext catalogs; converted to “.txt” via Python script
- TXT meta data
 - Tintable? Flipable? ...

Basic Bitmap Manipulation

- Flip / Mirror

```
for (y = 0; y < height; y++)  
{
```

```
    src.x = 0;  
    src.y = y;  
    src.w = width;  
    src.h = 1;
```



```
    dest.x = 0;  
    dest.y = height - y + 1;
```

```
    SDL_BlitSurface(dest_surf,  
                   &dest,  
                   src_surf,  
                   &src);
```

```
}
```

(Swap X/Y/width/height for mirror)

Basic Bitmap Manipulation

- Thumbnail

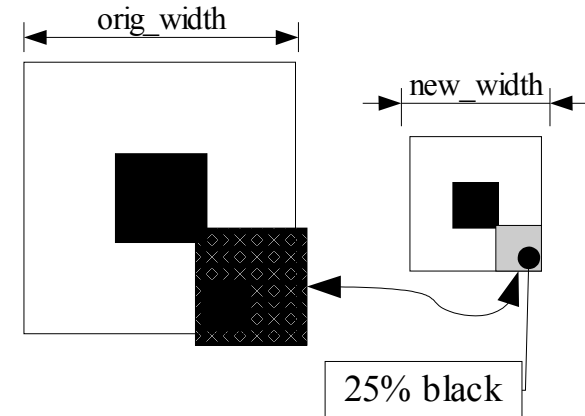
```
xscale = orig_width / new_width; // float  
yscale = orig_height / new_height; // float
```

```
for (y = 0; y < new_height; y++) {  
    for (x = 0; x < new_width; x++) {  
        tr = tg = tb = ta = 0;
```

```
        for (src_y = y * yscale; src_y < y * yscale + yscale; src_y++) {  
            for (src_x = x * xscale; src_x < x * xscale + xscale; src_x++)  
            {  
                /* Get RGBA values from (src_x, src_y) in src. surface;  
                 Add values to tr, tg, tb and ta */  
            }  
        }  
    }  
}
```

```
/* Divide tr, tg, tb and ta by the number of pixels captured */  
/* (xscale * yscale) */
```

```
/* Put new tr, tg, tb and ta values at (x, y) in dest. surface*/  
}  
}
```



Advanced Bitmap Manipulation

- Smudge – Crazy Albert Cahalan code

```
static double state[32][32][3];

i = 32*32;
rate = button_down ? 0.5 : 0.0;

while (i--) {
    iy = i>>5;
    ix = i&0x1f;

    // is it not on the circle of radius sqrt(120) at location 16,16?
    if ( (ix - 16) * (ix - 16) + (iy - 16) * (iy - 16) > 120)
        continue;

    // it is on the circle, so grab it
    SDL_GetRGB(getpixel_canvas(canvas, x+ix-16, y+iy-16), ..., &r, &g, &b);
    state[ix][iy][0] = rate * state[ix][iy][0] +
        (1.0-rate) * sRGB_to_linear_table[r];
    // Same for green and blue...

    // opacity 100% --> new data not blended w/ existing data
    putpixel(canvas, x + ix - 16, y + iy - 16,
        SDL_MapRGB(canvas->format, linear_to_sRGB(state[ix][iy][0]) ... );
}
```

Cvts 0-255 to 0.0 to 1.0

Advanced Bitmap Manipulation

- Tint – More crazy cool Albert code

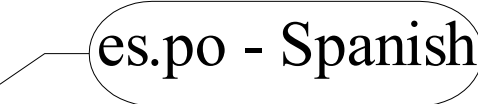
```
/* Grab original pixel... */
SDL_GetRGB(getpixel(src_surf, x, y), ..., &r, &g, &b);

/* We'll affect new color based on our eyes' perception of hue */

old = sRGB_to_linear_table[r] * 0.2126 +
      sRGB_to_linear_table[g] * 0.7152 +
      sRGB_to_linear_table[b] * 0.0722;

/* Draw in new color (rd,gd,bd), affected by orig. pixels (old) */
putpixel(dest_surf, x, y,
         SDL_MapRGB(...,
                    linear_to_sRGB(rd * old),
                    linear_to_sRGB(gd * old),
                    linear_to_sRGB(bd * old)));
```

Internationalization (“i18n”)

- Inside C code, wrap strings in calls to “gettext()”:
 - `strcpy(str, gettext(“Blue”))`
- Run “xgettext” console tool against source files to generate “tuxpaint.pot” 'template' file:
 - `msgid “Blue”`
`msgstr “”`
- Translators load “.pot” into poEdit, Kbabel or a plaintext editor and fill in the blanks, send to me:
 - `msgid “Blue”`
`msgstr “Azul”` 

Internationalization (“i18n”)

- Add Makefile target to “compile” new language:

```
trans/xy.mo: src/po/xy.po
             msgfmt -o trans/xy.mo src/po/xy.po
```

- Add Makefile cmds to install compiled “.mo” file:

```
cp trans/xy.mo /usr/share/locale/xy/LC_MESSAGES/tuxpaint.mo
```

- Add command-line & config file support:
(Tux Paint provides “--lang xylandian” options)

```
if (strcmp(argv[i], "--lang") == 0 && i < argv - 1)
    ... if (strcmp(argv[i + 1], "xylandia") == 0)
        putenv("LANGUAGE=xy_XY");
        putenv("LC_ALL=xy_XY");
```

Internationalization (“i18n”)

- Tell gettext to use the domain

```
bindtextdomain("tuxpaint", LOCALEDIR);

/* Old version of glibc does not have
   bind_textdomain_codeset() */

#if defined __GLIBC__ && __GLIBC__ == 2 &&
    __GLIBC_MINOR__ >=2 || __GLIBC__ > 2

    bind_textdomain_codeset("tuxpaint", "UTF-8");

#endif

textdomain("tuxpaint");
```

Printing

- On Windows & BeOS
 - SDL surface sent to their printing APIs
- On Linux and Mac OS X:
 - Originally SDL surface converted to PNG and sent to:
`popen("pngtopnm | pnmtops | lpr", "w");`
 - Now SDL surface converted to PostScript and sent to:
`popen("lpr", "w");`

Printing

- Hold [Alt] to get a print dialog
 - Windows – uses Windows API
 - BeOS – uses BeOS API
 - Mac OS X – uses Mac OS X API
 - Linux – calls an alternative print command (defaults to “kprinter”; can be overridden, as can “lpr”)
- Options are available to parents/teachers, to:
 - never allow print dialog (even if [Alt] is held), or
 - always show dialog (even if [Alt] is not held)

Printing

- PostScript printing:

```
fprintf(fi, "%%!PS-Adobe-3.0 EPSF-3.0\n");
fprintf(fi, "%%%Title: (%s)\n", fname);
strftime(buf, sizeof buf - 1, "%a %b %e %H:%M:%S %Y", localtime(&t));
fprintf(fi, "%%%CreationDate: (%s)\n", buf);
fprintf(fi, "%%%Creator: (Tux Paint " VER_VERSION ", " VER_DATE ")\n");
        //...etc. etc... (see src/tuxpaint.c)
fprintf(fi, "/DeviceRGB setcolorspace\n");
fprintf(fi, "<<\n");
fprintf(fi, "  /ImageType 1\n");
fprintf(fi, "  /Width width /Height height\n");
fprintf(fi, "  /BitsPerComponent 8\n");
fprintf(fi, "  /ImageMatrix [width 0 0 height neg 0 height]\n");
fprintf(fi, "  /Decode [0 1 0 1 0 1]\n");
fprintf(fi, "  /DataSource currentfile\n");
fprintf(fi, ">>\n");
fprintf(fi, "%%%BeginData: %u Binary Bytes\n", surf->w * surf->h * 3u);
fprintf(fi, "image\n");
        // Literally write every pixel's R/G/B value to "fi" stream... */
fprintf(fi, "\n");
fprintf(fi, "%%%EndData\n");
fprintf(fi, "grestore\n");
fprintf(fi, "showpage\n");
fprintf(fi, "%%%EOF\n");
```

Threaded Start-up (new in 0.9.15)

- Reduce start-up time of Tux Paint by only waiting for *required* data to load (UI elements & sounds)
- Uses `SDL_thread.h` on Windows, `fork()` on Linux
- Creates a “`socketpair()`” for communications
- Shares variable arrays, so `SDL_ttf` just loads into main Tux Paint process' memory
- Uses an “I'm done loading” boolean flag, if not done by the time user wants to use Text tool, they are forced to wait
- Code is still a bit... messy-looking

Project Hosting

- **SourceForge.net:**
 - CVS source code repository
 - Mailing lists (for developers, users & announcements)
 - Trackers: Bugs, Requests For Enhancement (RFE)
 - File storage / download mirrors
- **NewBreedSoftware.com** (*my website @ Sonic.net*)
 - Website – PHP code with translations via `<? include ?>`
 - PHP stored in SF CVS for easy access by translators
- **Tux4Kids**
- **Freenode.net**
 - IRC chat channel (“#tux4kids”, “#sdl”, etc.)

Project Management

- All volunteer-based (*I'd love to get paid for it!*)
- Coders & translators get CVS write access
(They need SourceForge accounts)
- Some i18n arrives via the “Translation Project”
(It's harder to keep these in sync., unfortunately!)
- Mailing lists used for discussing/arguing changes,
CVS commit logs
- SourceForge now used for bugs & to-do lists

What's Next?

- More improvements to work with school-specific setups:
 - Awkwardly-designed login procedures
 - Windows Terminal Server 2003
 - More tweaks for [K12]LTSP thin client labs
- Code cleanup:
 - Remove unused code
 - Split source into multiple files/modules

What's Next?

- “New” command UI improvements
 - Move 'Starter' images into a dialog accessed via “New”
 - Add colors to “New” dialog, to start with non-white canvas
- Stamp Categorization & UI improvements
 - ~~Some stamp contributors didn't follow my half-baked directory structure~~ (*cleaned up in November 2005*)
 - Don't list all 200+ stamps (and hopefully more!) all at once
 - Provide a few major categories that users can switch between, download separately, enable/disable via config

The End

Bill Kendrick

bill@newbreedsoftware.com

Tux Paint website

<http://tuxpaint.sf.net/>

Tux Paint SourceForge Project

(CVS, mailing lists, bug tracker, downloads)

<http://www.sf.net/projects/tuxpaint/>

Tux4Kids

<http://www.tux4kids.org/>