Co-maintaining cairo: cool community cool code

Behdad Esfahbod
besfahbo@redhat.com

Carl Worth
cworth@redhat.com
Intro
Rendering model
History
April 2002
Hi Keith,

I read up a bit on PostScript. It's the obvious source of the proposals you were making.

I like it much better than GL as a model for Xr. It's cleaner in general, (IMHO). And PS has larger overlap with the primitives we want in Xr, (eg. bezier curves as opposed to just triangle meshes).

So, here's what I'm thinking of so far. This sticks fairly close to PostScript, (but with an explicit rather than an implicit state object).

Feedback welcome,

-Carl
/* Opaque state structure */
typedef struct _XrState XrState;

/* Functions for manipulating state objects */

/* XXX: Do we want to add 'State' to any of these functions?
   eg. XrStateCreate, XrStateClone? */
XrState *XrCreate(void);
void XrDestroy(XrState *xrs);

void XrSave(XrState *xrs);
void XrRestore(XrState *xrs);

XrState *XrClone(XrState *xrs);

/* Modify state */
void XrSetPicture(XrState *xrs, XrPicture);
void XrSetColor(XrState *xrs, XrColor);

/* XXX: XrSetLineWidth, XrSetLineCap, XrSetLineJoin, XrSetDash, ... */

/* Path creation */

/* XXX: I'm thinking that it might make sense to do away with the
   notion of a "current path" in the state object and instead
   provide functions to manipulate an opaque XrPath object. This
   would add one more argument to XrStroke/XrFill, but it would
   unify support for PS "user paths" */

void XrNewPath(XrState *xrs);
void XrMoveTo(XrState *xrs, double x, double y);
void XrLineTo(XrState *xrs, double x, double y);
void XrClosePath(XrState *xrs);

/* XXX: XrRLineTo, XrArc, XrCurveTo, XrRCurveTo, ... */

/* Render current path */
void XrStroke(XrState *xrs);
void XrFill(XrState *xrs);
<table>
<thead>
<tr>
<th>Year</th>
<th>Committers</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002</td>
<td>2</td>
</tr>
<tr>
<td>2003</td>
<td>11</td>
</tr>
<tr>
<td>2004</td>
<td>12</td>
</tr>
<tr>
<td>2005</td>
<td>26</td>
</tr>
<tr>
<td>2006</td>
<td>54</td>
</tr>
<tr>
<td>2007</td>
<td>44</td>
</tr>
<tr>
<td>Year</td>
<td>Backend</td>
</tr>
<tr>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>2002-06</td>
<td>Xlib</td>
</tr>
<tr>
<td>2003-02</td>
<td>image</td>
</tr>
<tr>
<td>2003-10</td>
<td>PostScript</td>
</tr>
<tr>
<td>2004-04</td>
<td>XCB</td>
</tr>
<tr>
<td>2004-09</td>
<td>glitz</td>
</tr>
<tr>
<td>2005-01</td>
<td>Win32</td>
</tr>
<tr>
<td>2005-01</td>
<td>PDF</td>
</tr>
<tr>
<td>2005-01</td>
<td>Quartz</td>
</tr>
<tr>
<td>2005-12</td>
<td>SVG</td>
</tr>
<tr>
<td>2005-12</td>
<td>BeOS</td>
</tr>
<tr>
<td>2005-12</td>
<td>directfb</td>
</tr>
<tr>
<td>2006-09</td>
<td>OS/2</td>
</tr>
<tr>
<td>2007-02</td>
<td>Quartz (New!)</td>
</tr>
</tbody>
</table>
Bindings

C++
Common Lisp
D
Haskell
Java
.NET
Nickle
O'Caml
Perl
PHP
Python
Ruby
Scheme
Squeak
News
Mutex rework
Error-handling
Great malloc hunt
PDF hotness
Xlib work queue
Lessons
If you don't *git* it, you don't get it.

It's the Right Thing To Do.
Tiny commits

Earn points faster

APPLY NOW
Commit-access for everyone!
Tricking others into doing *their* work
Don't go AWOL
Why do we love cairo?
Why do you love cairo?