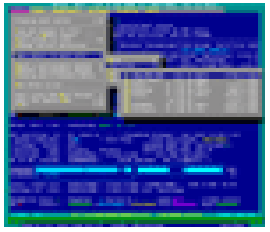
TxWindows, a multi platform text mode windowing library

Jan van Wijk

The TxWindows library for development of text mode applications using a windowing system including menus, dialogs and more ...

FSYS - Software



TxWin

Presentation contents

- What & Why, text-mode windowing
- Main features in Txwindows
- Some samples and demos
- Current shortcommings, future ...
- Availability and licencing

What is ...

- A text mode windowing system (UI)
  - A user interface based on the well-known principles of GUI's like Windows and OS2-PM, but operating entirely with text screen elements (ASCII, ANSI ...)
  - Uses UI elements like windows, buttons, lists, menus, entryfields; Is operated using keyboard and/or mouse.
  - As opposed to:
    - Graphical User Interface (GUI) environments
    - Simple STDIO based text applications that only use the keyboard and simple sequential output to the screen

Why text mode windowing

- Portable to many platforms
- Fast, even on old hardware
- Works in minimal environments like boot-diskettes (OS2, DOS, Linux ...)
- Appeals to commandline users

Main features

- Multi-platform, currently available on:
  - DOS, 32-bits, using a DOS-extender
  - OS2, 32-bits only, OS/2 2.x and later
  - Windows 32-bit, NT and later (no 9x)
  - Linux, should work on almost any distribution
- API much like OS2-PM or Windows
- Message based, extendible to a certain degree using custom window procedures
- Windows build dynamically, no resource-files
  (Instead, Widget lists can be used for easy dialog creation)

Message processing

- The main application function contains a message processing loop (like OS/2 PM):

```
while (txwGetMsg(  &qmsg))
{
   txwDispatchMsg( &qmsg);
}
```

- Dialogs have their own message loop, inside the 'txwDlgBox()' library function (these are modal dialogs in nature)

Familiar message names

- TxWindows uses familiar messages like:
  - TXWM_CREATE
  - TXWM_DESTROY
  - TXWM_CHAR
  - TXWM_COMMAND
  - TXWM_CLOSE
  - TXWM_QUIT
  - TXWM_HELP
  - TXWM_PAINT
  - TXWM_SETFOCUS
  - TXWM_MOVE
  - TXWM_SIZE
  - TXWM_CONTROL
  - TXWM_BUTTONDOWN
  - TXWM_USER

Familiar windowing functions

- TxWindows equivalents for many tasks:
  - txwSendMsg
  - txwPostMsg
  - txwCreateWindow
  - txwSetFocus
  - txwInvalidateWindow
  - txwQueryWindow
  - txwQueryWindowUShort
  - txwQueryWindowPos
  - txwSetWindowPos
  - txwDefWindowProc
  - txwDefDlgProc
  - txwDismissDlg
  - ...

Features, continued

- Includes common classes or controls like:
  - Frame or Canvas, text-window
  - Static text (output fields)
  - Text-viewer window (can be used for HELP)
  - Scrolling output window with status area
  - Buttons, including Check-box and radio-button
  - Entryfield, including history list popup
- Most standard behaviour implemented in default window-procedures in the library
  - Specific window procedures are mainly needed when controls in a dialog have mutual dependancies

Features, dialogs

- Includes a few standard dialogs:
  - Message-box, with one to four buttons (W*)
  - Prompt-box, to get simple single field input (W*)
  - File-Open and File-save-as dialogs (W*)
  - Directory picker dialog (W*)
  - Menubar dialog, with pulldown and sub menus
  - List-box, as submenu or standalone popup
  - Widget dialog, easy creation of custom dialogs (W*)
- The (W*) marked dialogs can be extended very easily using a list of Widget definitions
  - Any CONTROL class can be used in a widget list

Output handling

- Windows based, application specific
- 'printf' based, with output redirected to the large scroll-buffer window, or STDOUT and optional logfile (for LOG and TRACE).
  - The 'printf' based output is also available when no windowing is used at all, in that case the output will go the the STDOUT stream and optional logfile.
    It does support full ANSI support to allow cursor positioning and use of colors, either in the buffer or directly to the screen.

Argument and Option parsing

- Integrated parsing of program/command arguments and options (or switches)
- Allows nesting, and available throughout the whole application, not just in main()
- Has query functions like:
  - TxaArgCount
  - TxaArgValue
  - TxaGetOption
  - TxaOptionStr
  - TxaOptionNum
  - ...

Functional tracing

- Offers tracing of function ENTRY and EXIT, as well as parameters or other events.
  - Available inside the library itself, can be used by the application by means of a set of specific TRACE macros like 'ENTER()'
  - Tracing can be redirected to a file for analysis, displayed on a statusline or to the normal output stream
  - Start trace using a command or a hotkey

Non-windowing functionalty

- The libarary has several modules and functions NOT related to windowing:
  - Additional string manipulation, like wildcards
  - Directory and file itterator (SysFileTree like)
  - Standard command interpreter/executor
  - LZW compression functions
  - REXX or NATIVE script processing
  - CRC calculation functions (several :-)
  - Filesystem information functions
  - Filename manipulation including wildcards
  - ...

Current limitations

- Not all behaviour is easy to extend/overrule
- Not modular enough (minimum code-size)
- Linux version not truly terminal-independant
  (but works OK in Console, KDE and Gnome :-)
- Minor issues:
  - No mouse support in Linux version yet
  - Unreliable mouse-cursor in OS2 full-screen

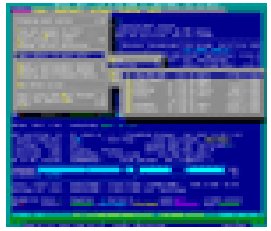
Availability and licencing

- Version 1.0 of TxWindows is available now
  - Interested developers can download the current version
    For more info and support, contact: txwin@dfsee.com
  - Distribution will be in the form of ZIP files for the library, samples, sources and the development environment
  - Source-control will be made available through NetLabs
- Licensing will be very similar to LGPL
  - Source updates should flow back to the community
  - Commercial applications OK, but no derived libraries

TxWindows, a multi platform text mode windowing library

Questions ?

FSYS - Software



TxWin