Current status of Win32 Gdk implementation

Bertrand Bellenot - bellbert@scopus.ch

Features (recall)

- Same environment on every system :
  - Same look and feel on every platform.
- Simplify the code maintenance :
  - No need to care about a « windows specific code ».
- Simplify functionality extension :
  - No need to implement the code twice, once for windows and once for other OS.
  - Only use TVirtualX.

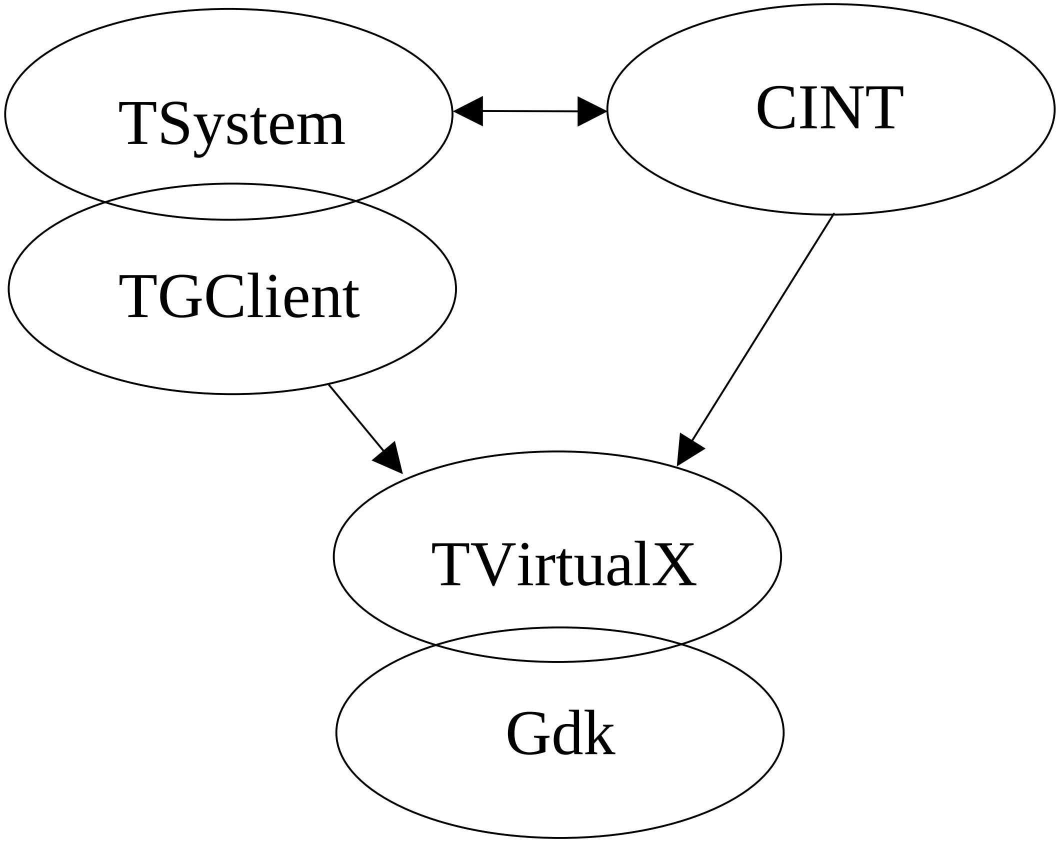
Actual Status (recall)

- The actual code uses a modified version of gdk and glib, the GIMP low-level libraries ported on win32. In practice, this means that we only need to link with gdk.lib, glib.lib and iconv.dll as additional libraries (hopefully less in the future). These libraries are under LGPL, so there are no licensing issues in using and distributing them.
- As original version of gdk was not doing everything needed by root (as font orientation!), I did have to slightly modify the original code.

Points fixed since last year

- Some characters were not displayed. ✓
- Some problems with icon’s transparency. ✓
- The event handling was not perfect. ✓
- OpenGL was not working. ✓

Events handling architecture (actual)



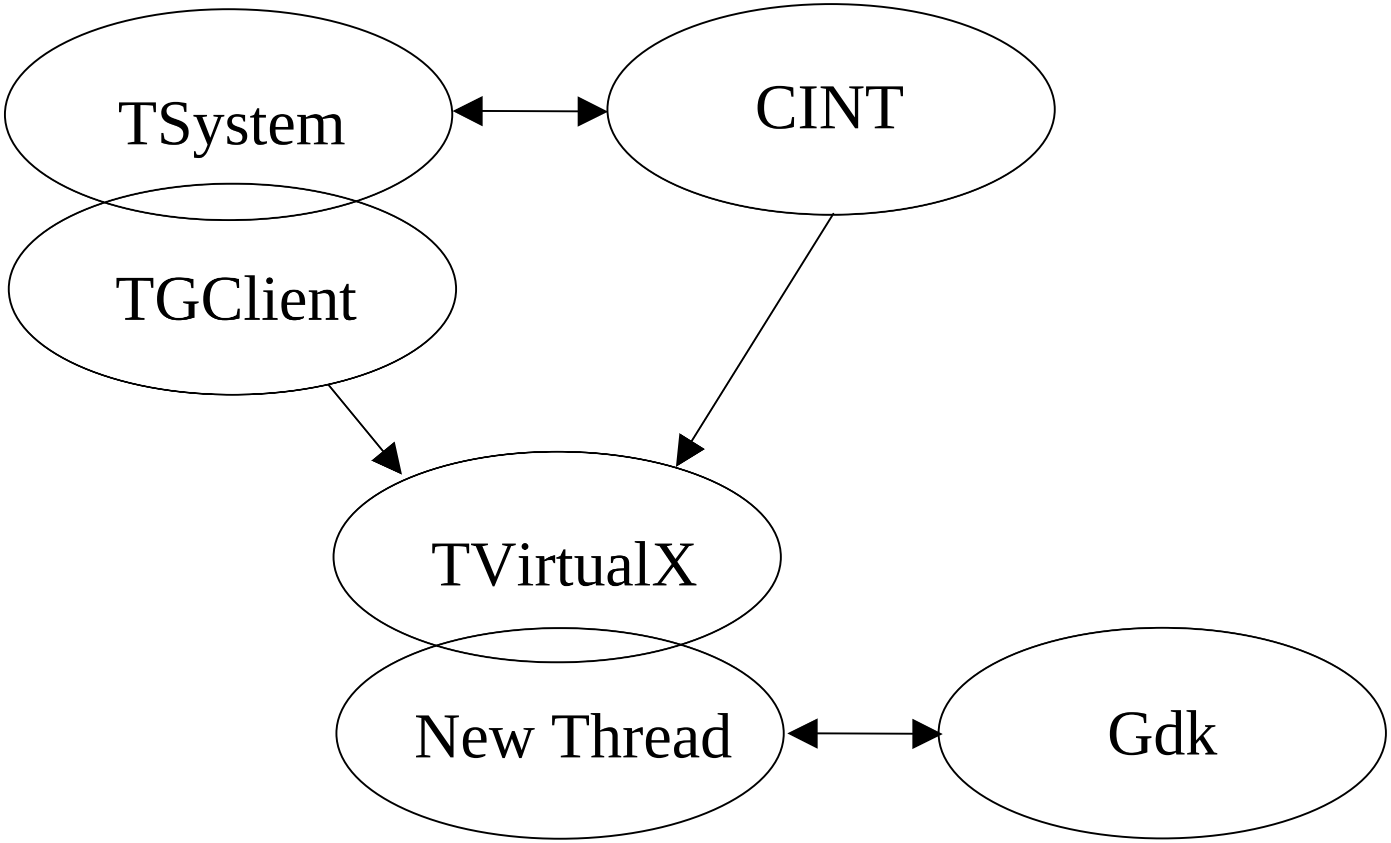
Threads issue

- From gdk developper FAQ :
- Without some major restructuring in GDK-Win32, I don't think there is any chance that GTK+ would work, in general, in a multi-threaded app, with different threads accessing windows created by other threads.
- One problem is that each thread in Windows have its own message queue. GDK-Win32 currently uses just one "message pump" in the main thread. It will never see messages for windows created by other threads.

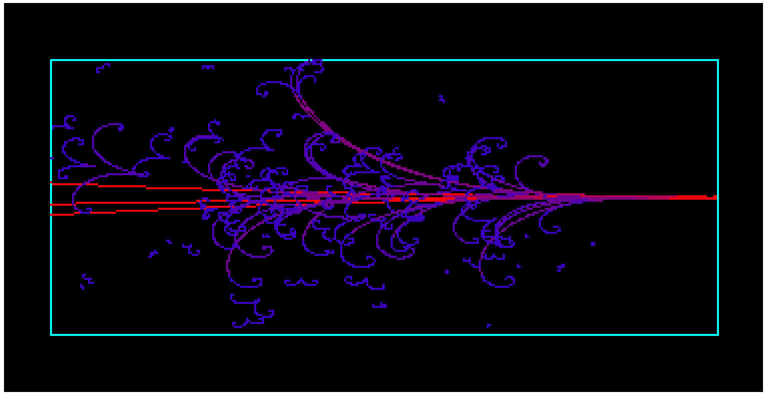
Threads issue

- As gdk is not thread safe, I had to create a separate thread from within the gdk calls are made.
- This implementation is working (more or less), but there are some problems :
  - It is slower than the native win32 implementation.
  - The CPU usage is always 100% (due to the way event loop is done actually).
- To improve the performance, several layers in gdk-glib should be replaced by a native win32 implementation.

Events handling architecture (under development)



Applications running with actual implementation



particle shower simulation
and event display

Alcan process data viewer

To Do List (Sorted by Priority)

- Improve threads.
- Implement a new “win32 native” implementation.
- ...

Annexes

- Needs.
  - If someone knows well ROOT, win32 and threads, his/her help is welcome.
  - More feedback from win32 users...
- Questions - Infos.
  - Feel free to contact me at my private e-mail addresses : bellbert@scopus.ch.

More informations

Status and progress reports will be on the Root Win32 GUI web page :

http://root.cern.ch/root/win32progress/Win32GUI.html

Thank’s to ROOT team for their great job, their support and for this workshop.

And thank you everybody for your patience.

Questions

You only arrive at the right answer after making all possible mistakes...