G4/vis/ToolsSG The ZB drivers

G4 Sapporo 2023 workshop

Guy Barrand, CNRS/IN2P3/IJCLab

vis/ToolsSG recap

- A vis driver introduced in 2021 based on a scene graph logic developed at LAL (now IJCLab) since 2010, and for which the code is now in g4tools (under tools/sg).
- Good part of the code is on C++ standard libs.
- Some rather light code in toolx to bind to various renderer and windowing systems, today: OpenGL, X11, Xt, Windows, Qt (Qt5 and Qt6).
- Few code in visualization/ToolsSG: only the "glue" to the general/generic G4/vis system.
- It permits some plotting (see G4-Rennes slides).
- Try to keep some free "academic way" to do visualisation.

vis/ToolsSG rendering

- The screen rendering is virtualised at the level of the scene graphs, so we can introduce various ones.
- From G4/10.x we have a GL-ES screen renderer working with the windowing systems supported by G4/vis, then the drivers: TSG_[QT,X11,XT,WINDOWS]_GLES.
- In G4/11.1 had been introduced an offscreen renderer: TSG_OFFSCREEN. It is based on a standalone g4tools/z-buffer implementation. See G4-Rennes slides.

TSG_<windowing>_ZB

- In fact we can use also the g4tools/z-buffer to render in a screen window! All windowing systems permit to create a window/widget for "pixmap" rendering.
- Since done 100% on CPU, would it be fast enough for interactive visualisation?
- Experience show that on recent laptops it is workable! (in particular the Apple "M" ones). Then in G4/11.2 we propose the drivers: TSG_[QT,X11,XT,WINDOWS]_ZB
- Since not based on any external graphics package, they could be built by default, as soon as a windowing system is choosen.

TSG_<windowing>_ZB (2)

- But do not expect too much about “rendering features” of this driver. It is not OpenGL! For example there is no antialiasing, triangle edging.
- But for what we ask primarily in G4/vis; draw points, lines, triangles in 3D, it does the job. (We have transparency!).

- A user support fallback: since not on OpenGL, a TSG_ZB driver may help to debug “nasty graphics user configurations” seen often in the vis forum (not adapted graphics card, visualising from a virtual machine, etc…).

Else

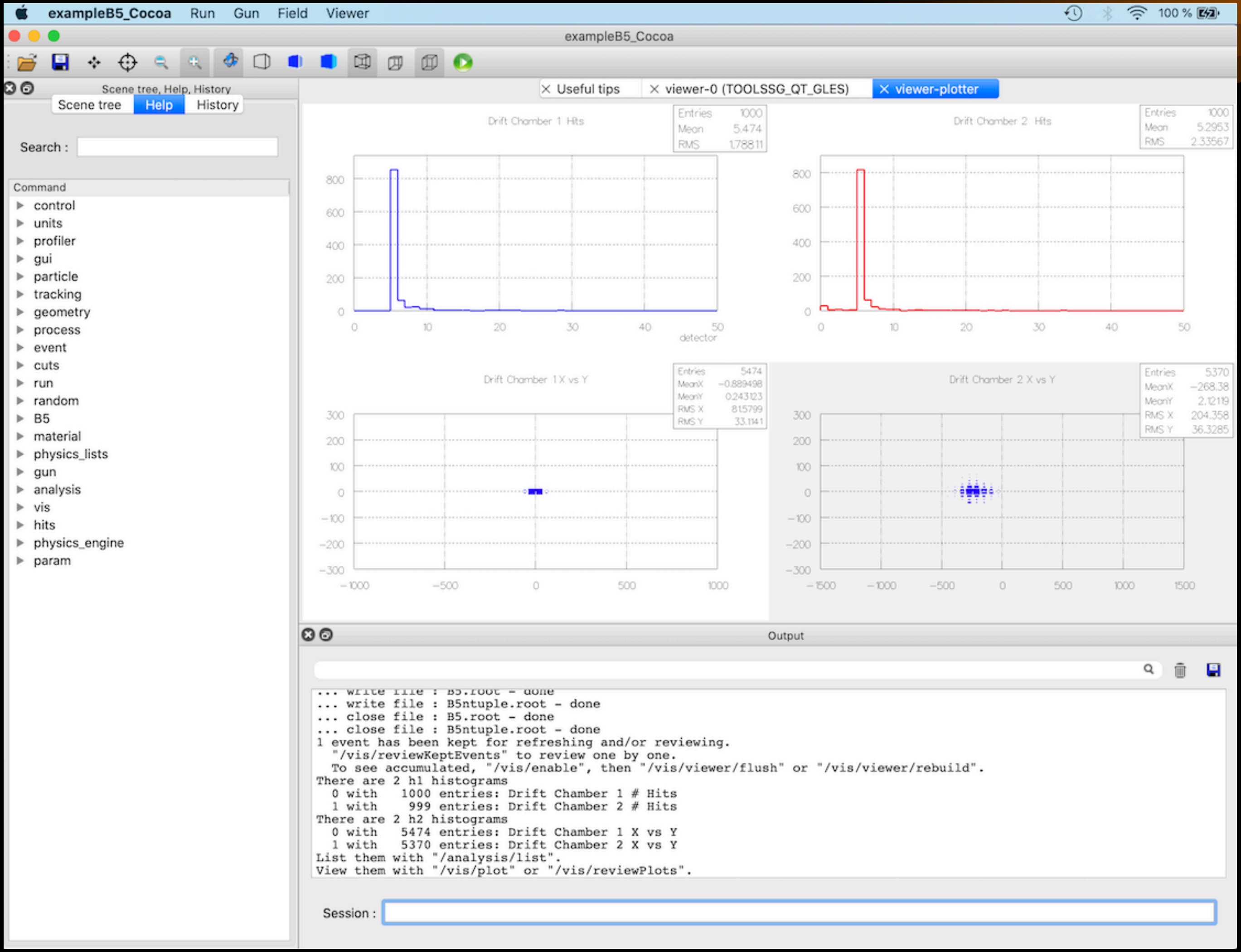
- We handle Qt6.
- For the TSG_<windowing>_GLES drivers, based on the OpenGL/2.x API, we saw on Macs run time clashes with VTK which is based on OpenGL/3.x API. Next year, may be good to move the TSG_GLES drivers to OpenGL/3.x…
- Various other improvements in the job list: 3D plotting, have similar rendering than the G4/vis/OGL drivers, same popup menu…

Backup slides (seen at Rennes)

vis/ToolsSG plotting

- tools/sg contains a “plotter node” (then based on the tools/ sg scene graph logic).
- Already used in G4/analysis for “batch plotting”.
- In G4/vis, had been introduced the G4Plotter model logic to be able to bind and activate this g4tools plotting from the G4/vis system and then also from the G4 command system.
- The G4/vis/plotting knows the histos in G4/analysis.
- (Not so easy to find the right way to connect all these!).
- See examples/basic/B5 for an example.

vis/ToolsSG plotting (2)



```
... write file : B5.root - done
... write file : B5ntuple.root - done
... close file : B5.root - done
... close file : B5ntuple.root - done
1 event has been kept for refreshing and/or reviewing.
  "/vis/reviewKeptEvents" to review one by one.
  To see accumulated, "/vis/enable", then "/vis/viewer/flush" or "/vis/viewer/rebuild".
There are 2 h1 histograms
  0 with   1000 entries: Drift Chamber 1 # Hits
  1 with    999 entries: Drift Chamber 2 # Hits
There are 2 h2 histograms
  0 with   5474 entries: Drift Chamber 1 X vs Y
  1 with   5370 entries: Drift Chamber 2 X vs Y
List them with "/analysis/list".
View them with "/vis/plot" or "/vis/reviewPlots".
```

vis/ToolsSG plotting (3)

- Today only 2D “flat” histo plotting, but more could come soon… (tools::sg::plotter can do 3D “lego” plotting and plots also functions).
- Today fully available with the ToolsSG driver, but within G4/vis, the logic is so (in particular the way to bind the G4/analysis histos) that other vis drivers “plotting able” (VTK?) may bring their plotting within the same architecture.

TSG_OFFSCREEN

- Then a new "sub driver": TSG_OFFSCREEN (beside TSG_[QT,X11,XT,WINDOWS]_GLES).
- To produce views (.ps, .png, .jpeg) straight from pure "batch", without having to tie to some Windowing system.
- Purely on C++ standard libs (then no X11, Windows, Qt, OpenGL around). It uses tools/sg/gl2ps_action but also the tools/sg/zb_action, a zbuffer renderer already in g4tools.
- It uses tools/gl2ps, but also new tools/fpng, toojpeg to write at the png and jpeg file formats.
- Fully thread safe (including gl2ps and the png, jpeg writers).
- g4> /vis/open TSG_OFFSCREEN
- Code is here, I hope to submit a MR soon…