EsbRootView / 2.0.0
More physics

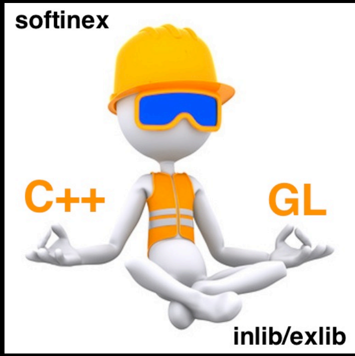
Zagreb ESSnuSB/EuroNuNet annual meeting

22 October 2019

Guy Barrand, CNRS/IN2P3/LAL

EsbRootView / guiding ideas (1)

- Have an event display able to run natively on all nice interactive devices that we have in hands today, by exploiting as much as possible local graphics capabilities of them.
- I have the graphics technology to do that for Linux/X11, Windows/Win32, macOS/Cocoa, iOS and Android.
- C++, local GL-ES and a scene graph logic of my own (strongly inspired by the great OpenInventor).
- Thesaurus of code and expertise accumulated for long, now on github under the generic name « softinex ».
- (Used also in G4/g4tools for doing IO and plotting).



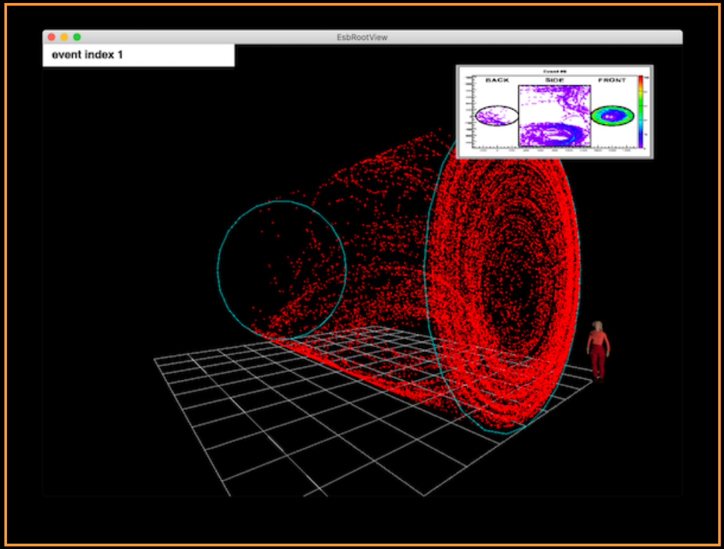
EsbRootView / guiding ideas (2)

- Data access : today existing various HEP frameworks/stacks, handling detector and event models (and the critical IO), because though/targeted for the batch (and then Linuxes), are not ported natively on the « interactive » operating systems.
- (iOS and Android are science fiction for them). Which is a sad fact for anyone interested in HEP, interactivity and visualization ☹.
- (This will not change before long).
- BUT, I can read detector and event ROOT files in a highly and light portable way. (softinex/inlib/rroot code) ☺

Use inlib/rroot over EsbRoot files and the softinex graphics classes to build EsbRootView.

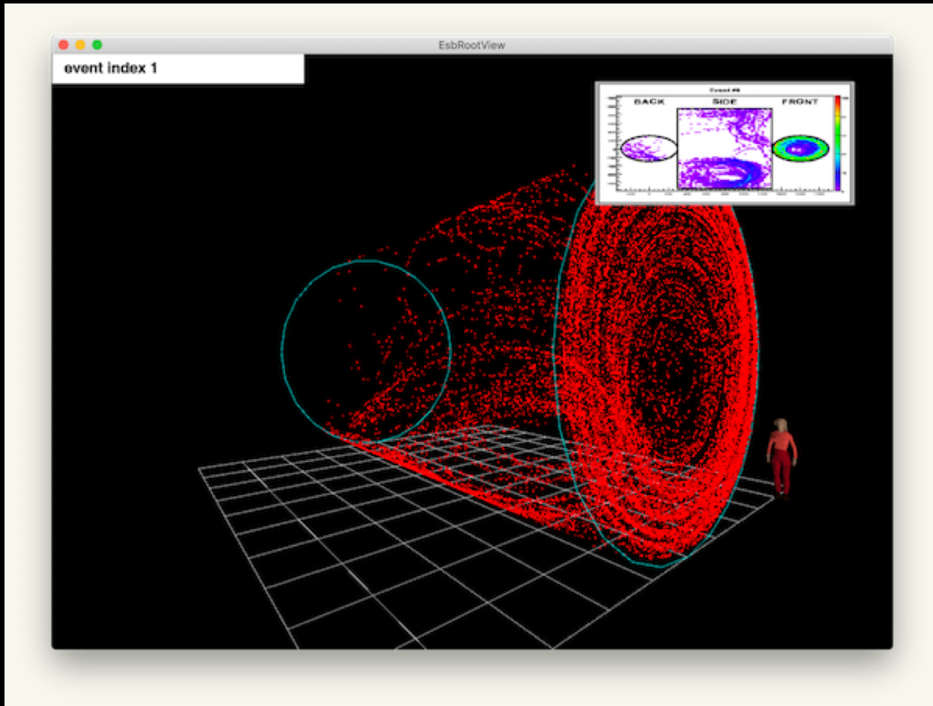
EsbRootView / 1.0.0

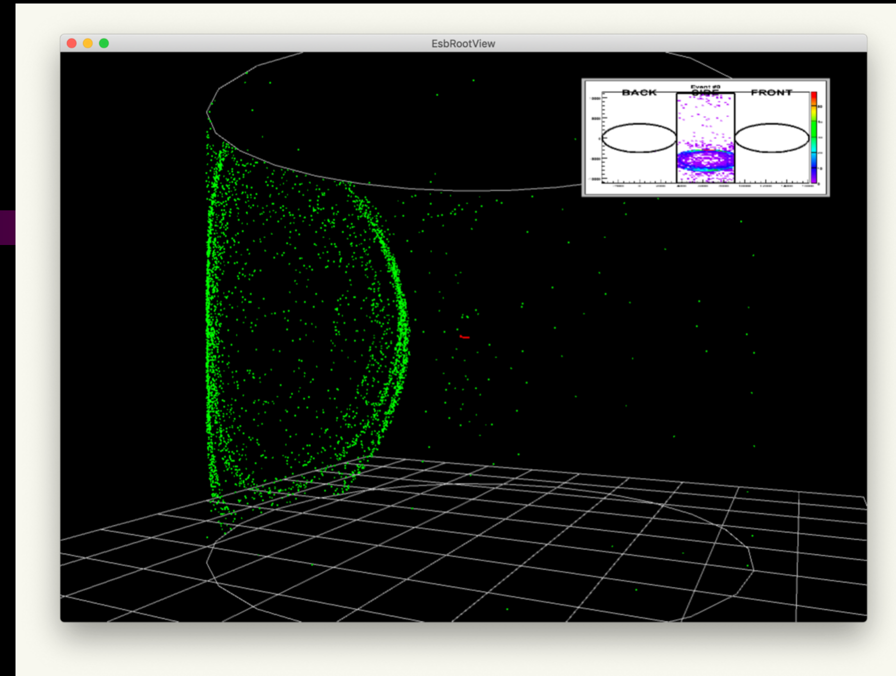
- Proof of concept/demonstrator that it is feasible.
- Read the geo_full.root and evetest.root of first release of EsbRoot.
- Show the “wc“ cylinder and WCDetectorPoints only.
- Released May/2019 on github/gbarrand.

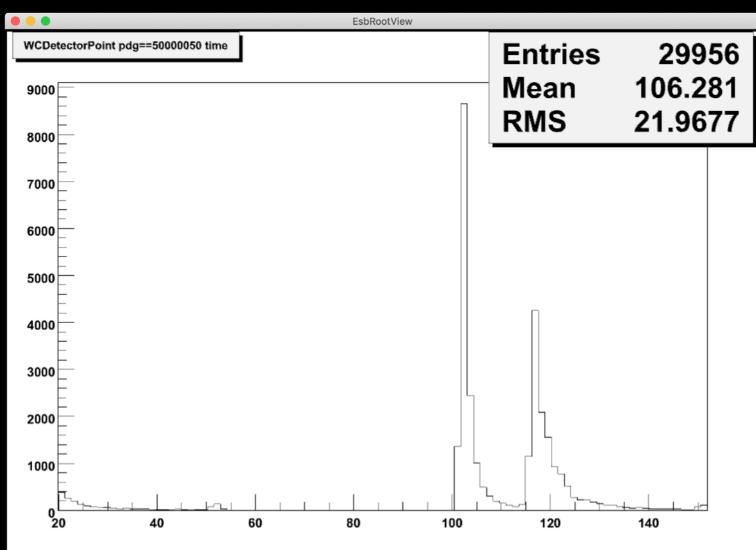


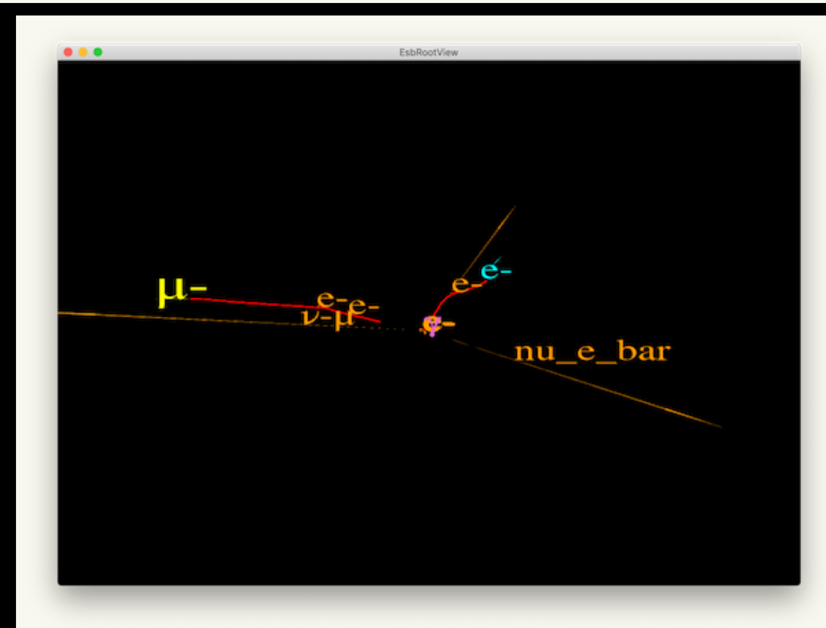
EsbRootView / 2.0.0/ More physics

- Released end September 2019.
- Now available from : https://gbarrand.github.io
- Can cover neard, fard, fgd setups with the same program.
- MCTrack point and « arrow » representations.
- WCDetectorPoint, FgdDectorPoint and FgdHit : point rep.
- FgdHit : solid and wire-frame cube representations.
- Evolution in time for MCTrack/t, [WC,Fgd]DetectorPoint/time.
- A « cut/filter » mechanism that permits to have a fine tuning of what we want to see: it helps a lot in understanding an event.
- A bash-like scripting to customize startup and event scenes (see web pages).

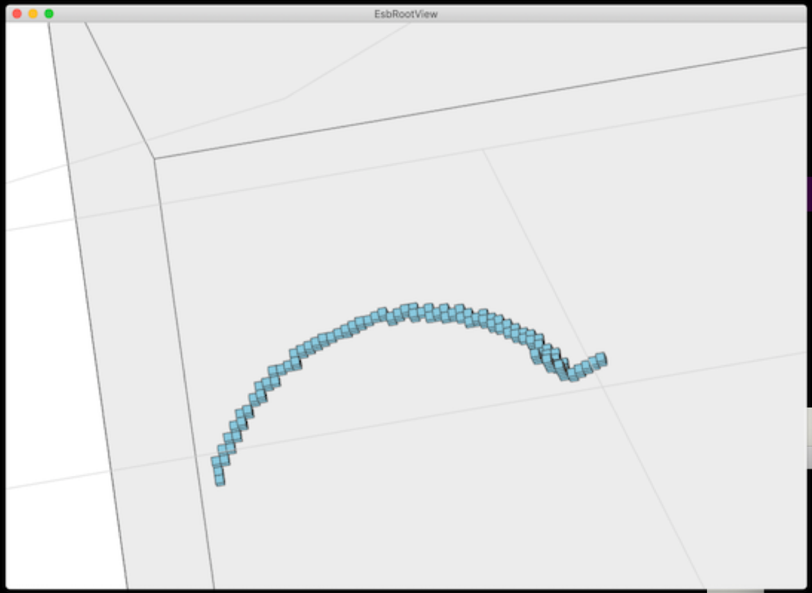

EsbRootView
event index 1
BACK
SIDE
FRONT

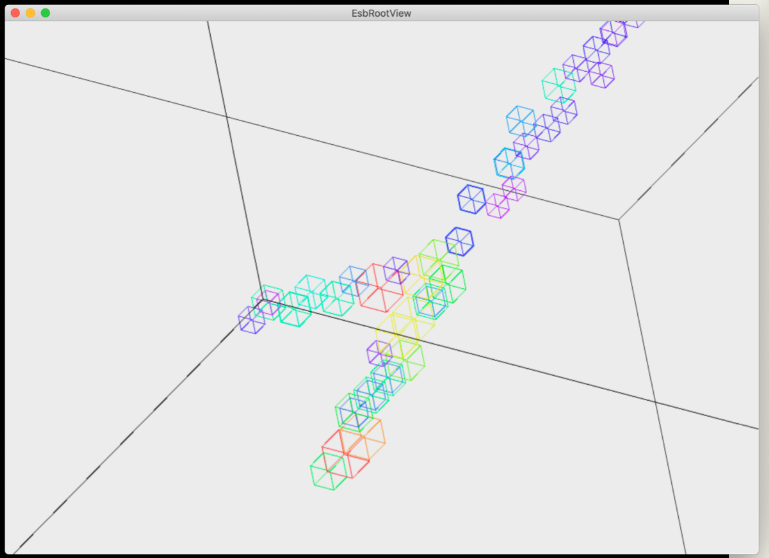

EsbRootView
BACK
SIDE
FRONT

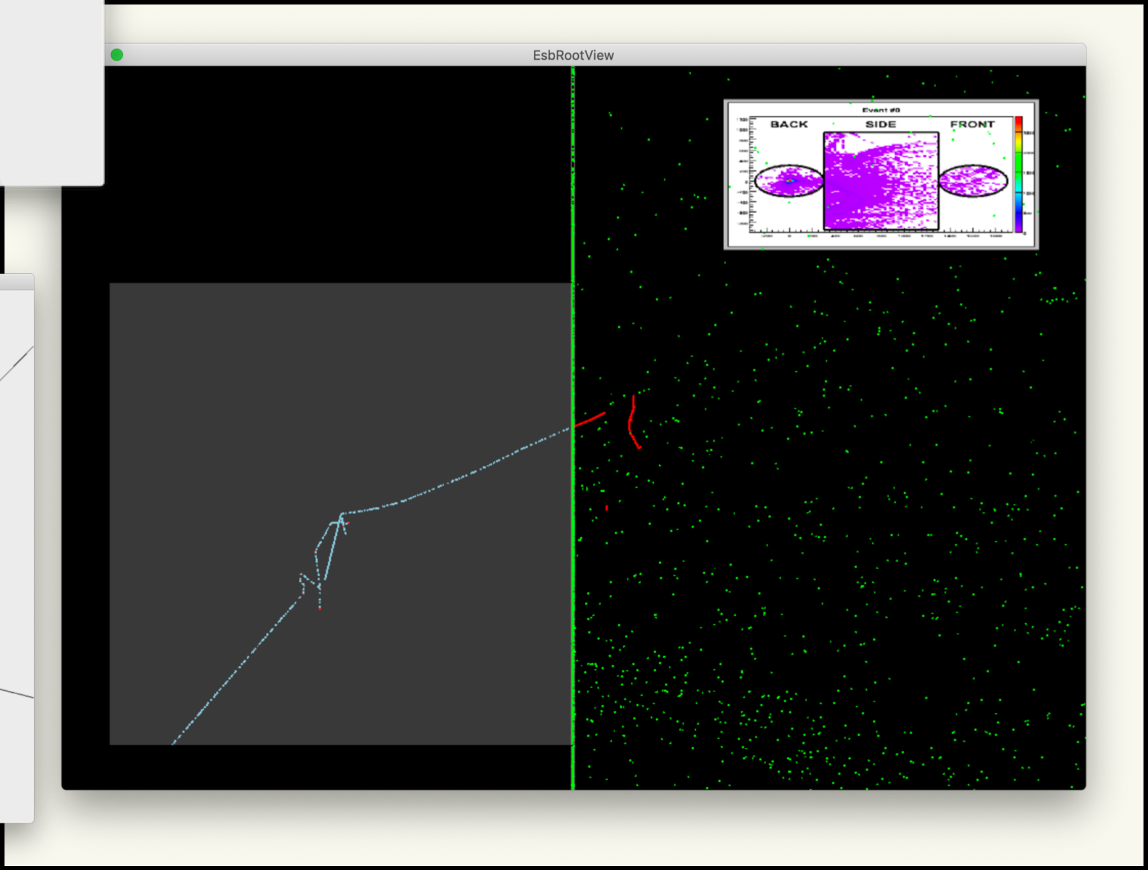

EsbRootView
WCDetectorPoint pdg==50000050 time
Entries 29956
Mean 106.281
RMS 21.9677
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
20
40
60
80
100
120
140


EsbRootView
μ-
e-
e-e-
ν-μ
e-
e-
e-
nu_e_bar

Fgd points and hits







Event deployment in time

- See the deployment in time of MCTracks and DetectorPoints.
- See some videos under the Gallery section of the web pages...
- Obviously, this feature can help for outreach...

- Seeing, for example a muon decay in such detectors, could reinforce the intuition of a newcomer in particle physics (students, engineers).

Scripting, -terminal, -cut

- Bash like scripting (then a flat learning curve).
- -terminal mode (with same history, completion, help logic as bash).
- Some accessor & evaluator logic over the event model that permits to pass cut/filter on data:

  event_vis MCTrack –cut=t<50 –color=red

  event_vis MCTrack –cut=t>50 –color=blue

  event_print MCTrack –cut=pdg==11 x y z E

  event_plot WCDetectorPoint -cut=pdg==50000050 -filling=p
- (-cut=<expression> available on all event model commands: event_vis, event_stats, event_print, event_plot).
- See web pages for more...

Next…

- More doc and customisation examples.
- Cone deployment animation.
- More event model items ? More representations ?

- Script the GUI (so that someone can easily create its own buttons).
- A first WebAssembly version ?
- ☹ Apple deprecated in June/2018 its OpenGL in favour of Metal on iOS and macOS. A problem for a lot of people in science (including Geant4/vis). Beside doing a sitting at Cupertino, I have to find time to write a « Metal renderer ».
- Deposit the app on GooglePlay and the Apple iOS/macOS app stores. I need an agreement of ESSnuSB for that (because the event and detector models are not mine).

Conclusions

- We are on good track to do something nice (and probably never done in HEP up so far).
- For the moment no technical resistance at the level of the IO (which is a critical point in the whole story).
- It would deserve a submission at some next CHEP conference.
- It gives a lot of ideas, at the level of the physics but also on software engineering to continue to do really interesting things.
- See you for the 3.0.0 ☺ !?…