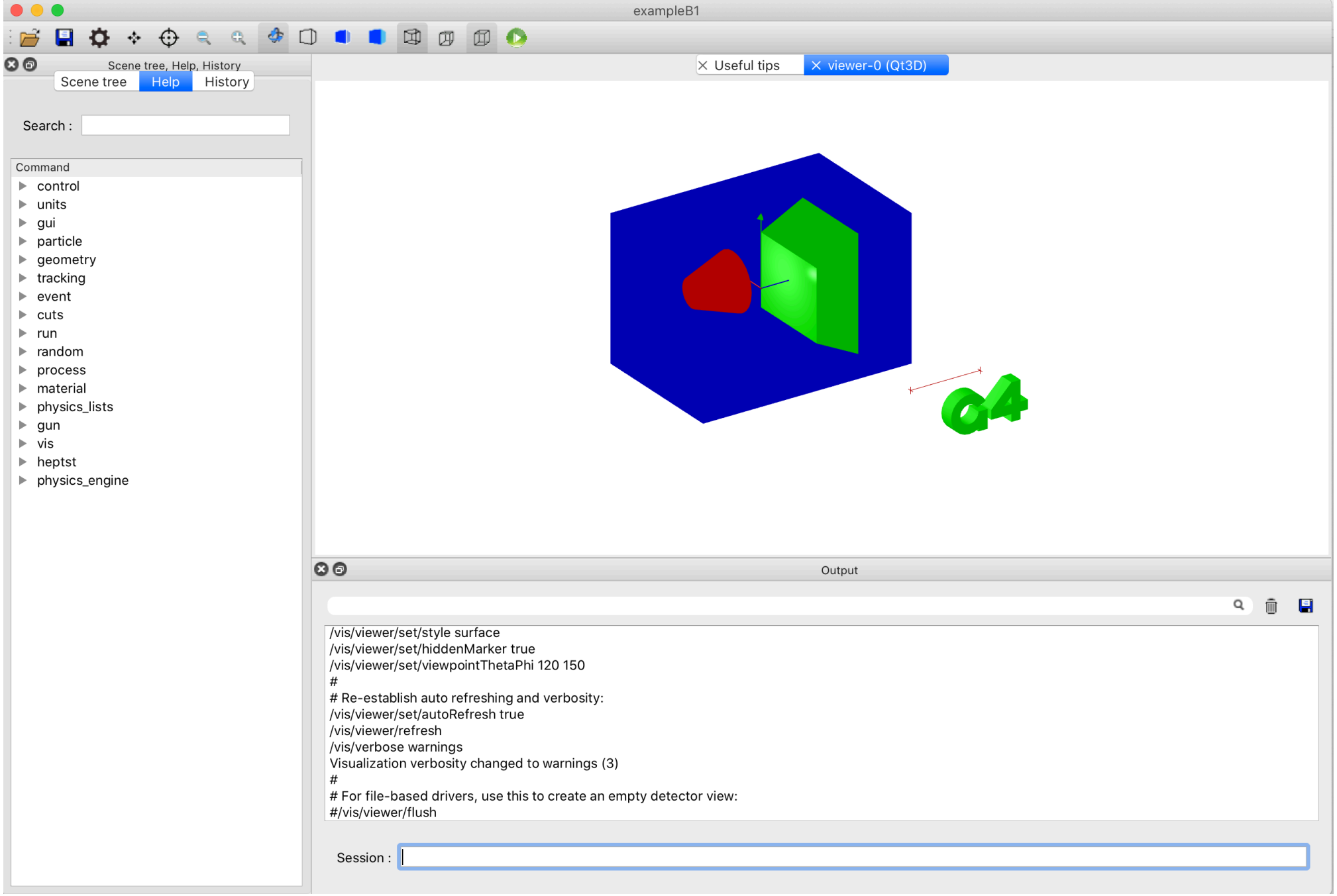
A Qt3D visualisation driver for Geant4

Qt3D

- Qt3D is a new graphical interface within the Qt project
  - Qt is free to open source projects
    - You have to register
- From https://www.qt.io/blog/2016/06/16/introducing-qt-3d:
  - "The Qt3DRender module is a high-level interface to hardware accelerated graphics. At present Qt 3D uses an OpenGL backend but we have left the door open to be able to support more modern APIs such as Vulkan, Metal and DirectX 12 in the future."
- Qt3D is still "under development"

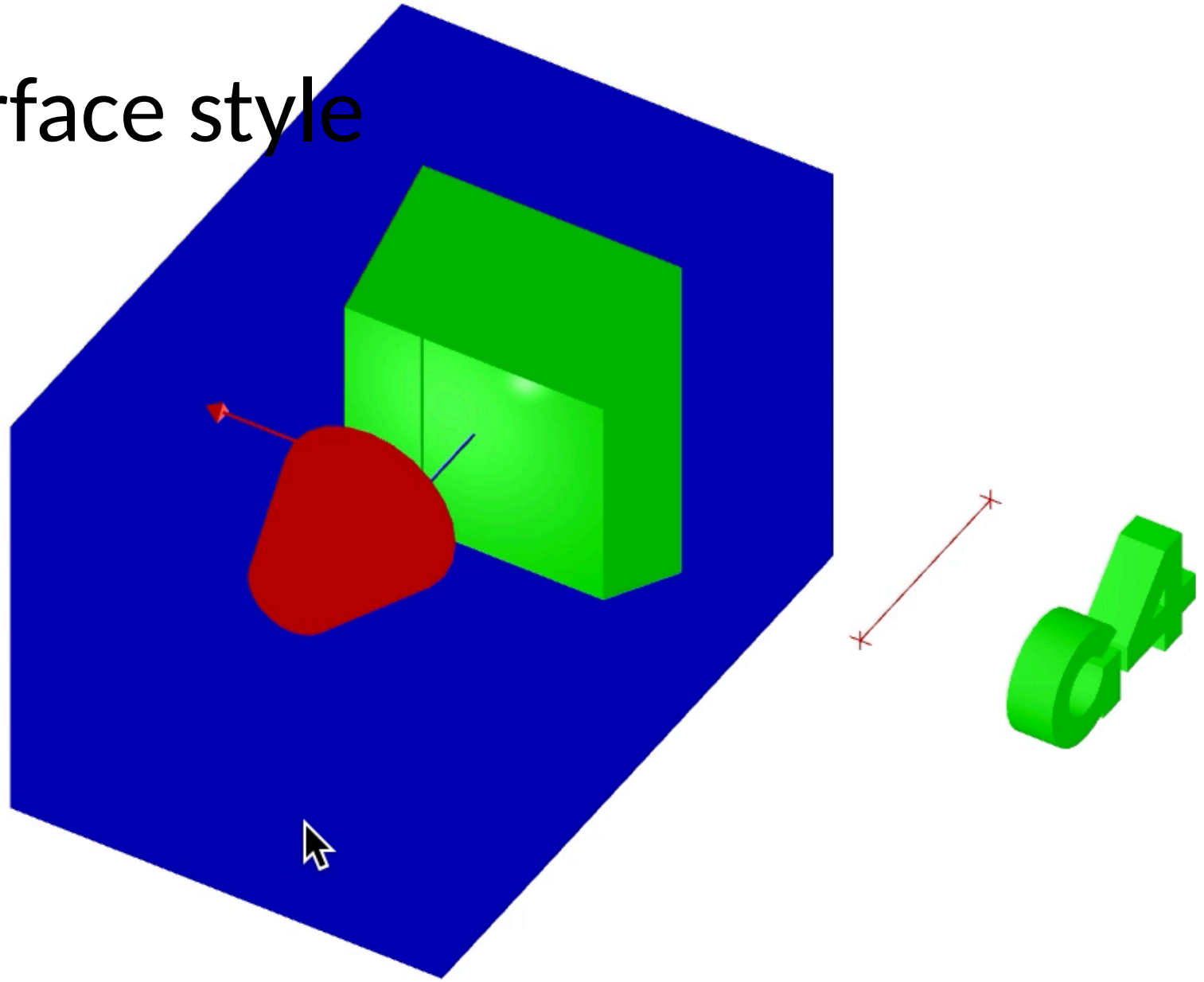
Experience so far

- Qt3D is pretty “low-level”
  - Documentation sparse; programming tough
  - It does not seem to have the very low level control of OpenGL
    - For example, I don’t see how to change line width
    - Or switch z-buffering off
    - But I guess it has to go for the highest common factor of all systems
- Geant4 Qt3D driver
  - Wireframe and surface drawing modes look OK
    - Hidden edge is weird
  - It doesn’t seem to do transparency very well
  - Trajectory drawing only works in sequential mode
    - I have not implemented changing threads yet
    - Markers (e.g., trajectory points) not yet implemented
  - Performance is good
  - In my personal repository, together with some examples
    - Until adopted you have to point to the G4visQt3D library and explicitly register it with the vis manager

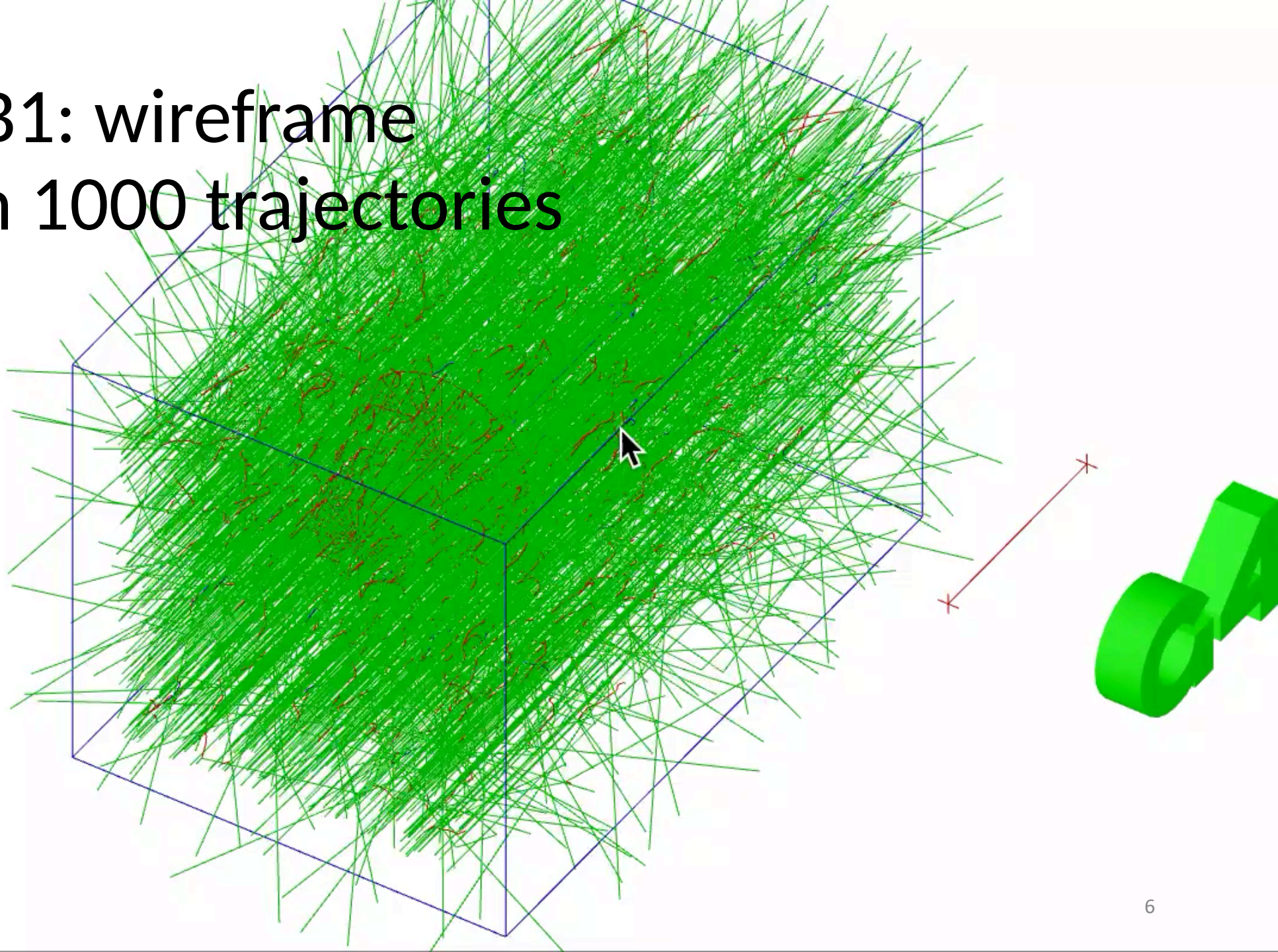

exampleB1
Scene tree, Help, History
Scene tree
Help
History
Search :
Command
control
units
gui
particle
geometry
tracking
event
cuts
run
random
process
material
physics_lists
gun
vis
heptst
physics_engine
Useful tips
viewer-0 (Qt3D)
Output
/vis/viewer/set/style surface
/vis/viewer/set/hiddenMarker true
/vis/viewer/set/viewpointThetaPhi 120 150
#
Re-establish auto refreshing and verbosity:
/vis/viewer/set/autoRefresh true
/vis/viewer/refresh
/vis/verbose warnings
Visualization verbosity changed to warnings (3)
#
For file-based drivers, use this to create an empty detector view:
#/vis/viewer/flush
Session :

exampleB1: surface style

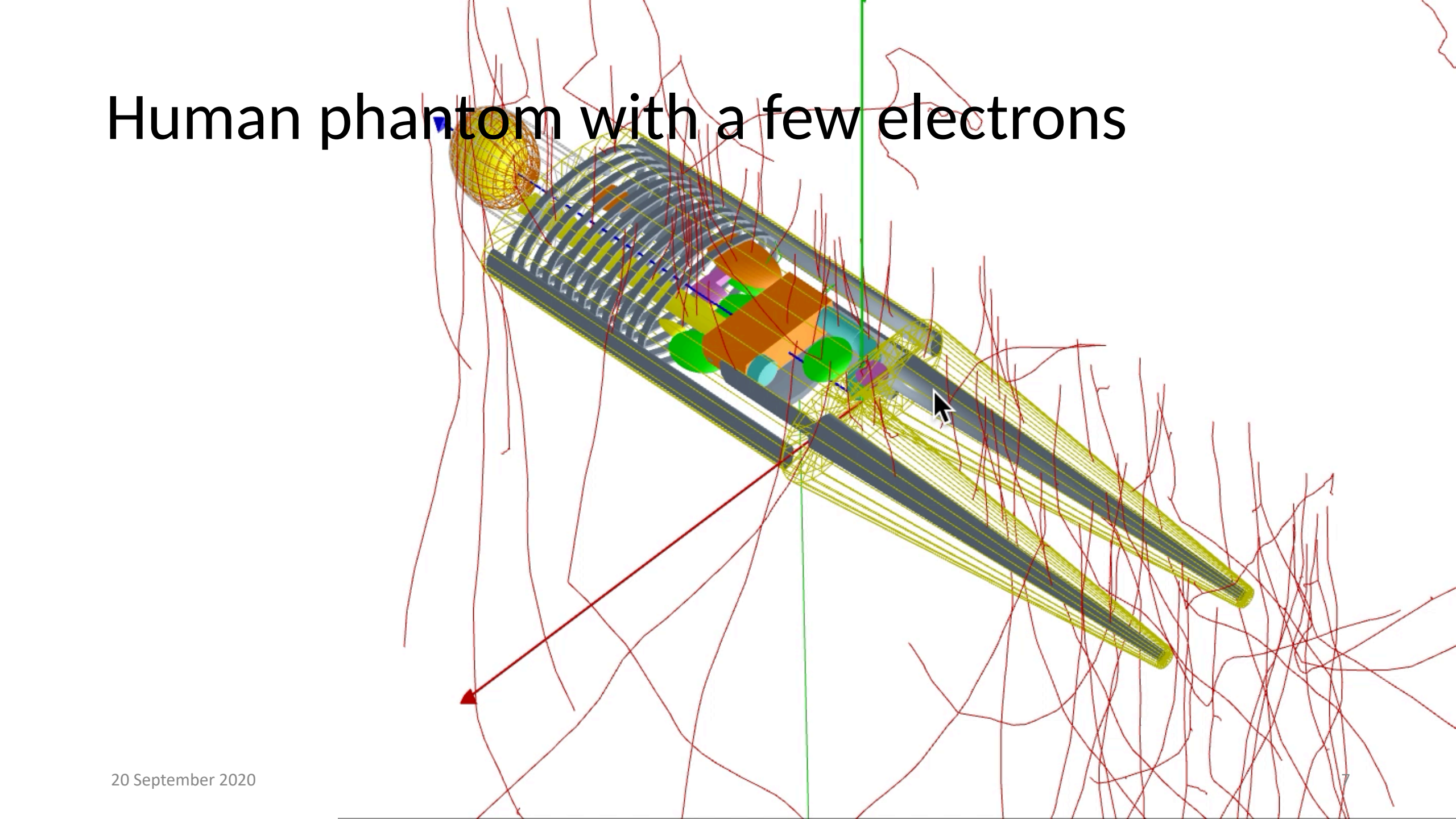
The blue “envelope” is supposed to be semi-transparent but it’s more like “see-through”



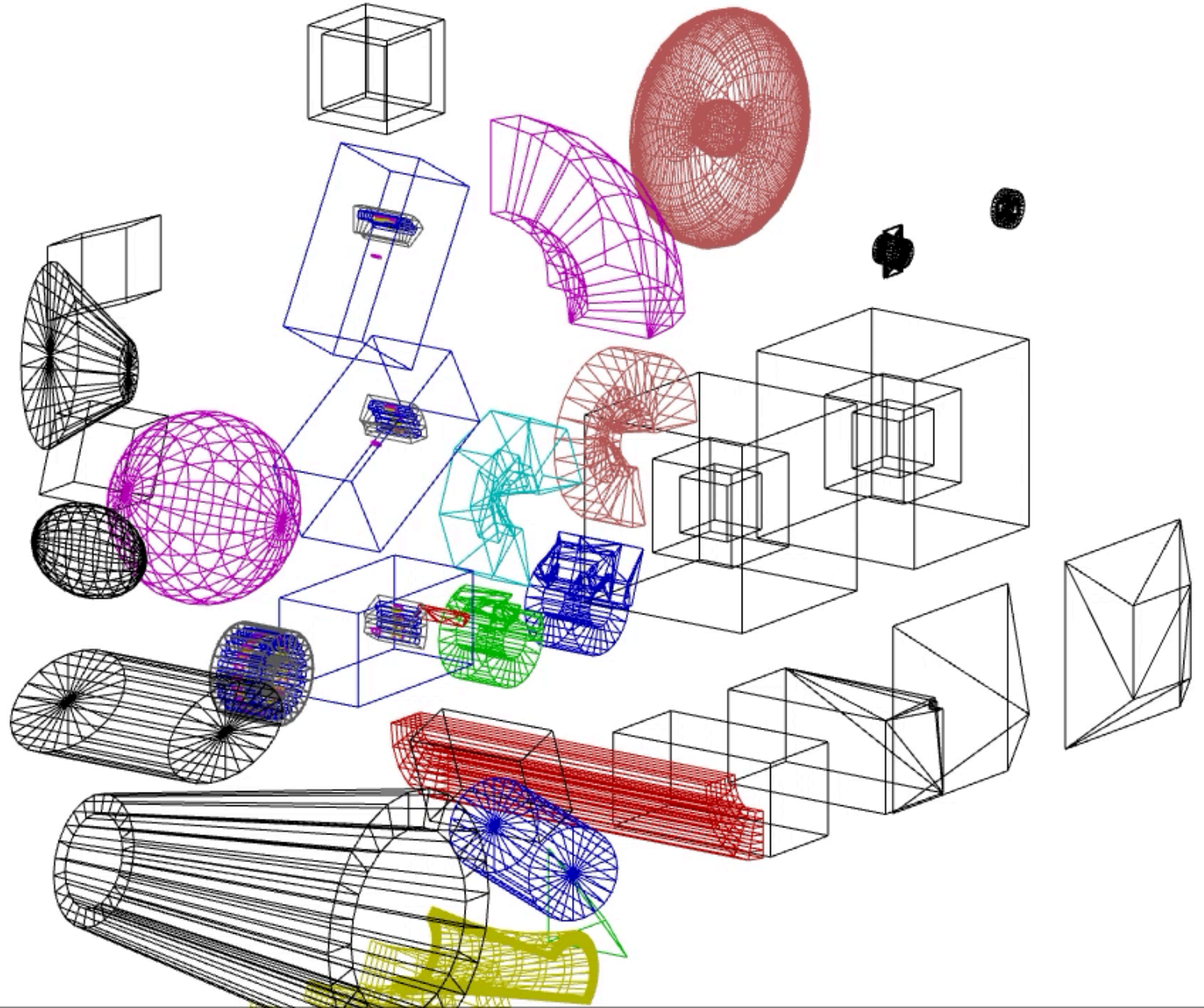
exampleB1: wireframe style with 1000 trajectories



Human phantom with a few electrons



test202



Summary

- A promising new driver
- Offers independence from evolution of graphics interfaces on all platforms