

New (and some old) Features of the Geant4 Visualisation System 2018

John Allison

Geant4 Associates International Ltd
and
The University of Manchester

Contents

- Tips
- New commands
 - /vis/drawLogicalVolume
 - /vis/drawOnlyToBeKeptEvents
 - /vis/viewer/colourByDensity
 - /vis/viewer/set/timeWindow/
- Old commands whose range of applicability has been extended
 - /vis/list (to see many new available colours)
 - /vis/set/touchable and /vis/touchable/set/
 - /vis/viewer/save and /vis/viewer/interpolate

Tips

- To get advice and information:
 - /vis/verbose confirmations
 - The default is “warnings”
 - Simple graded message scheme - digit or string (1st character defines):
 - 0) quiet, // Nothing is printed.
 - 1) startup, // Startup and endup messages are printed...
 - 2) errors, // ...and errors...
 - 3) warnings, // ...and warnings...
 - 4) confirmations, // ...and confirming messages...
 - 5) parameters, // ...and parameters of scenes and views...
 - 6) all // ...and everything available.
- To see commands echoed, and commands invoked by other commands:
 - /control/verbose 2

/vis/drawLogicalVolume

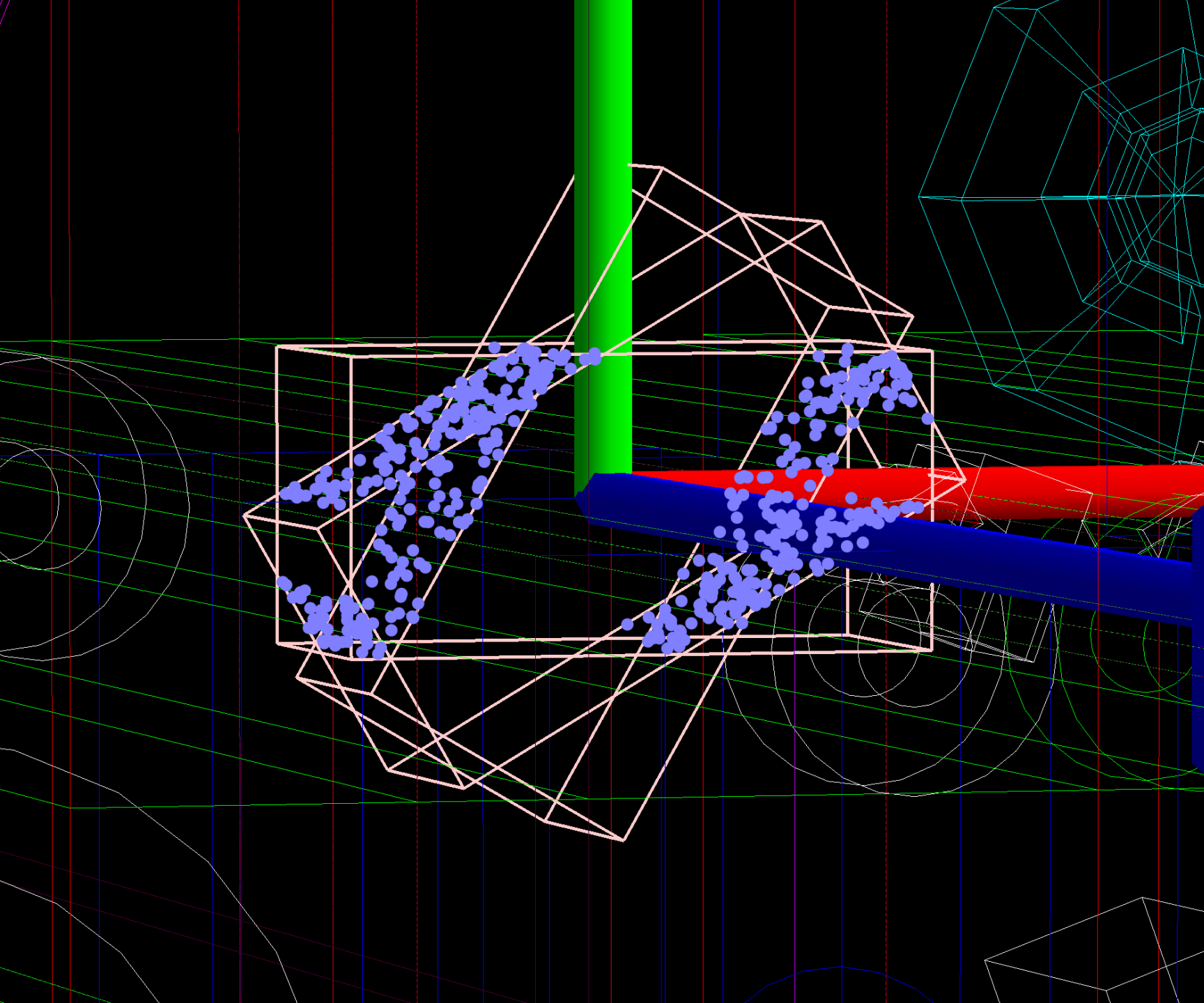
- Uses /vis/scene/add/logicalVolume
- (Synonymous with /vis/specify)
 - (Which was intended to mimic GEANT3's DSPEC.)
- Adds a logical volume to the current scene, shows boolean components (if any), voxels (if any), readout geometry (if any), local axes and overlaps (if any), under control of the appropriate flag.
- Note: voxels are not constructed until start of run - "/run/beamOn". (For voxels without a run, "/run/beamOn 0".)
- At present only *prints* overlaps. *Drawing* is "in discussion".

test202: an assembly of different shapes—a testbed for vis.

```
/vis/drawLogicalVolume expHall_L
```

Shows geometry voxels—red and green boxes—the voxelisation algorithm has not thought it necessary to divide in z, which would be in blue.

Note pink volumes and pale blue dots in centre—see zoom in next slide.



/vis/drawLogocalVolume (contd.)

Shows overlapping volumes (pink 5-pixel-thick lines) and sampled points of overlap (pale blue 20-pixel-diameter circles). (Still in discussion—hopefully for 10.5.)

The command also generates corresponding printing:

```
Checking overlaps for volume calo_phys ...  
----- WWWWW ----- G4Exception-START -----  
*** G4Exception : GeomVol1002  
issued by : G4PVPlacement::CheckOverlaps()  
Overlap with volume already placed !  
Overlap is detected for volume calo_phys:0  
with calo_phys:1 volume's  
local point (146.394,492.829,470.635), overlapping by  
at least: 7.17063 mm  
NOTE: Reached maximum fixed number -1- of overlaps  
reports for this volume !  
*** This is just a warning message. ***  
----- WWWWW ----- G4Exception-END -----
```

etc.

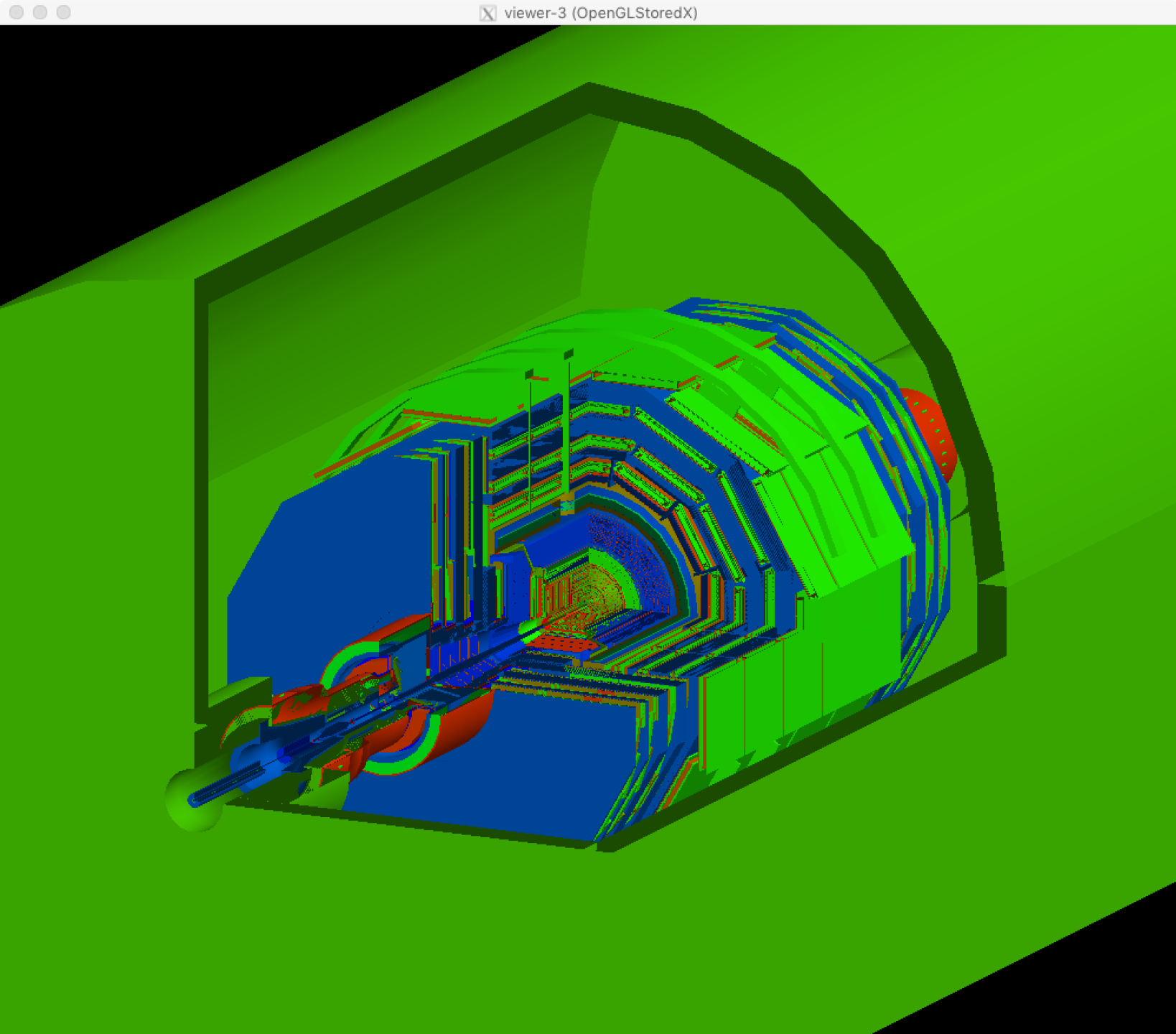
/vis/drawOnlyToBeKeptEvents

- Useful if you would like to see only some special events
- This command requests the vis manager to draw *during a run* only those events that have been designated "to be kept" by the user with `G4EventManager::GetEventManager()->KeepTheCurrentEvent()` or with `"/event/keepCurrentEvent"`. For example
 - ```
if (<some criterion>) {
 G4EventManager::GetEventManager()->KeepTheCurrentEvent();
}
```
- Alternatively, as before, you may `"/vis/disable"`, then *at end of run*
  - `/vis/enable`
  - `/vis/reviewKeptEvents`

## /vis/viewer/colourByDensity

- This asks the viewer to draw volumes that have no user-specified vis attributes in a colour determined by some algorithm that depends on density.
- Useful for geometries from GDML, which does not allow the specification of vis attributes.
- Currently only a simple algorithm is available:
  - If a volume has no vis attributes, colour it by density.
  - Provide algorithm number, e.g., "1" (or "0" to switch off).
  - Then a unit of density, e.g., "g/cm<sup>3</sup>".
  - Then parameters for the algorithm assumed to be densities in that unit.
  - Algorithm 1: Simple algorithm takes 3 parameters: d0, d1 and d2.
  - Volumes with density < d0 are invisible.
  - Volumes with d0 <= density < d1 have colour on range red->green.
  - Volumes with d1 <= density < d2 have colour on range green->blue.
  - Volumes with density > d2 are blue.
- See next slide for an example of CMS read from a GDML file.





```
$ load_gdml cms.gdml
```

```
..
```

```
/vis/viewer/set/viewpointThetaPhi 30. 30.
```

```
/vis/viewer/zoomTo 25
```

```
/vis/viewer/clearCutawayPlanes
```

```
/vis/viewer/addCutawayPlane 0 0 0 m 0 0 -1
```

```
/vis/viewer/addCutawayPlane 0 0 0 m -1 0 0
```

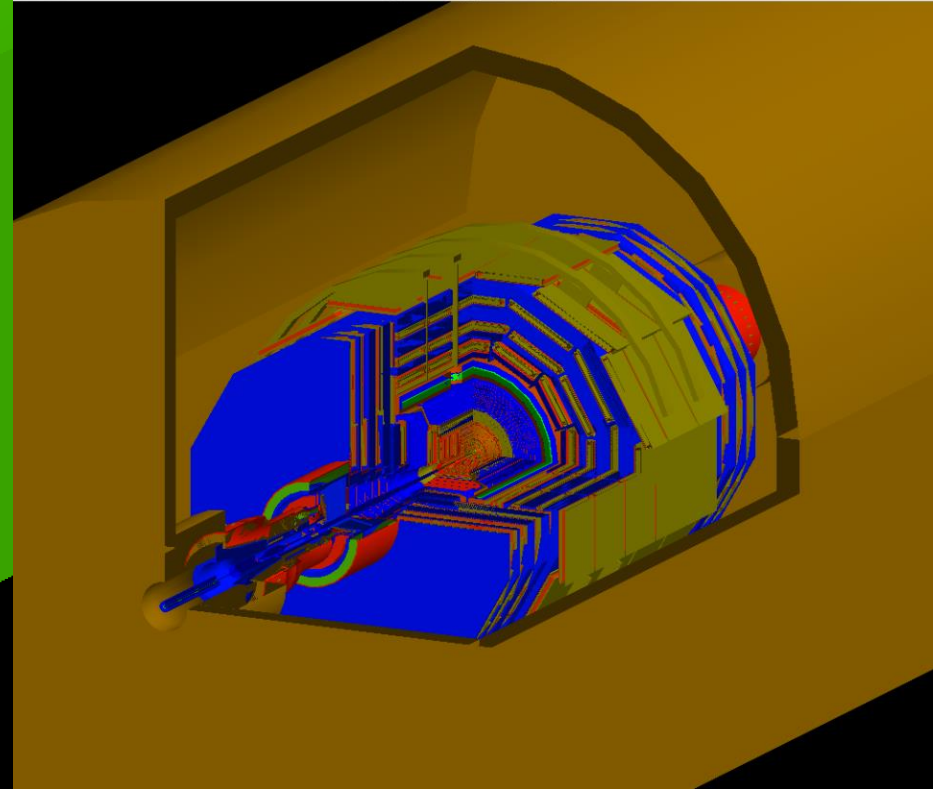
```
/vis/viewer/addCutawayPlane 0 0 0 m 0 -1 0
```

```
/vis/viewer/set/style surface
```

```
/vis/viewer/colourByDensity 1 g/cm3 .5 3 10 # left
```

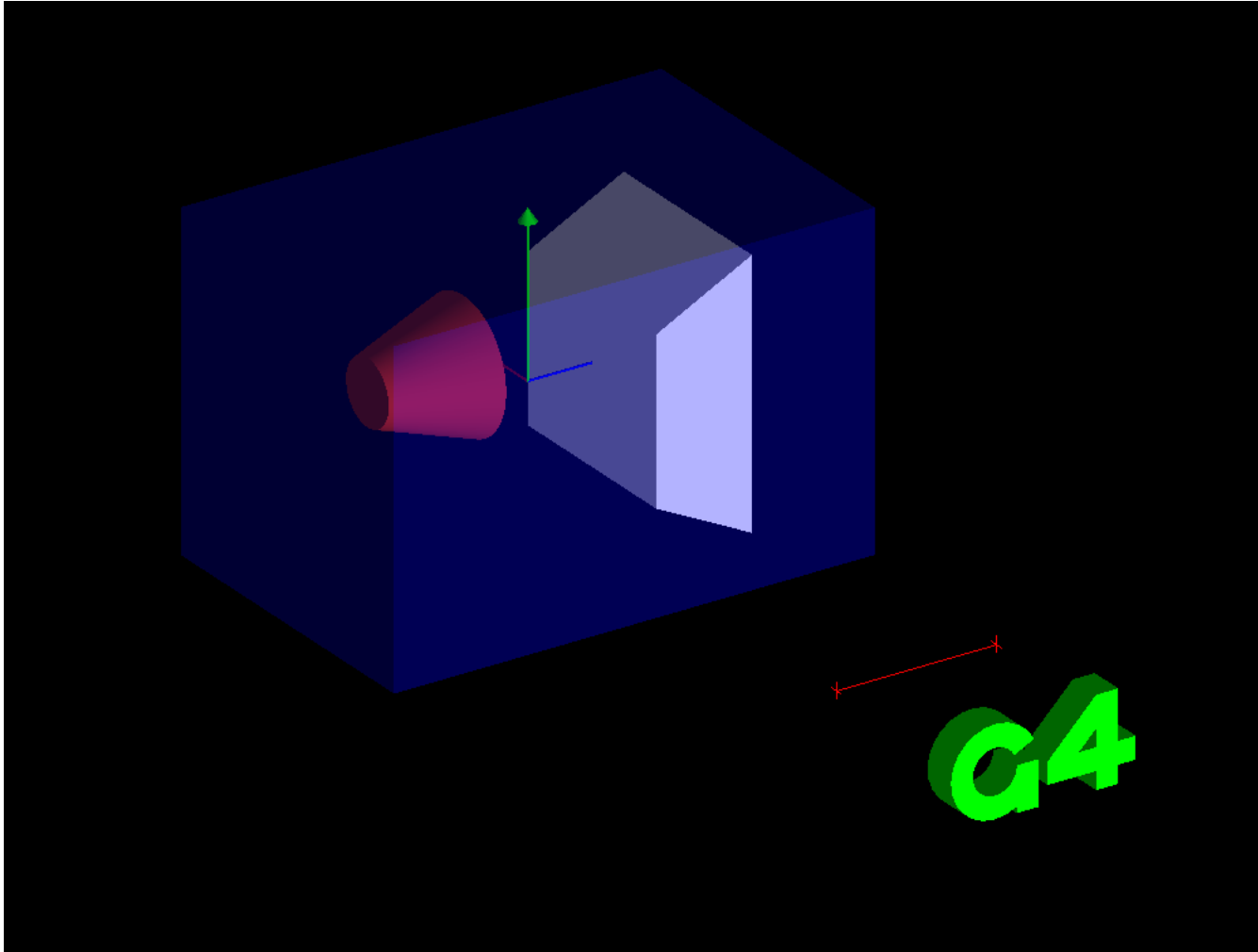
```
/vis/viewer/colourByDensity 1 g/cm3 .5 5 8 # below
```

viewer-3 (OpenGLStoredX)



# /vis/set/touchable and /vis/touchable/set/

- Information is stored in the view parameters (the commands do not change the geometry).
  - This means it can be saved and passed from view to view.
  - (This is in contrast to /vis/geometry/ commands, which add to or change vis attributes of G4LogicalVolume objects in the geometry tree).
- E.g., from exampleB1:
  - `/vis/drawTree`  
"World":0 / "World"  
"Envelope":0 / "Envelope"  
"Shape1":0 / "Shape1"  
"Shape2":0 / "Shape2"
  - `/vis/set/touchable World 0 Envelope 0 Shape1 0`
  - Perhaps there could be an option in drawTree that lists the touchables in a form that could be copied and pasted into /vis/set/touchable?
- Then
  - `/vis/touchable/set/colour firebrick`
- To see available text-specified colours:
  - Some /vis commands (optionally) take a string to specify colour (courtesy Guy Barrand's g4tools)
  - `/vis/list`  
Available colours:  
aquamarine, black, blue, blueviolet, brown, cadetblue, coral, cornflowerblue, cyan, darkgreen, darkolivegreen, darkorchid, darkslateblue, darkslategrey, darkturquoise, dimgrey, firebrick, forestgreen, gold, goldenrod, gray, green, greenyellow, grey, indianred, khaki, lightblue, lightgrey, lightsteelblue, limegreen, magenta, maroon, mediumaquamarine, mediumblue, mediumorchid, mediumseagreen, mediumslateblue, mediumspringgreen, mediumturquoise, mediumvioletred, midnightblue, navy, navyblue, orange, orangered, orchid, palegreen, pink, plum, red, salmon, seagreen, sienna, skyblue, slateblue, springgreen, steelblue, tan, thistle, turquoise, violet, violetred, wheat, white, yellow, yellowgreen



From exampleB1:

```
/vis/drawTree
```

```
"World":0 / "World"
```

```
 "Envelope":0 / "Envelope"
```

```
 "Shape1":0 / "Shape1"
```

```
 "Shape2":0 / "Shape2"
```

```
/vis/set/touchable World 0 Envelope 0 Shape1 0
```

```
/vis/touchable/set/colour firebrick
```

## /vis/viewer/set/timeWindow/...

- (Previously /vis/ogl/set/..., now DEPRECATED. Parameters have been moved from OpenGL to G4ViewParameters, so that they are applicable, in principle, to all viewers that implement them.)
- This sets a time window for drawing for the current viewer.
  - Default  $-\infty < t < \infty$ .
- Every drawable object also has a time range of validity (part of the vis attributes).
  - Default  $-\infty < t < \infty$ .
- If these time intervals overlap the object is drawn.
- Trajectories may be time-sliced. For example
  - /vis/scene/add/trajectories rich
  - /vis/modeling/trajectories/drawByCharge-0/default/setTimeSliceInterval 0.01 ns
- You can draw in a loop—see examples/basic/B4/macros/visTutor.
  - /control/loop visTutor/exN03Vis13.loop endTime 0 0.7 0.001
  - Then in visTutor/exN03Vis13.loop:
    - /vis/viewer/set/timeWindow/endTime {endTime} ns 0.1 ns
    - Put /vis/ogl/export in the loop and you get hundreds of pdf files (or other chosen format) to make a video.
- Or you can use /vis/viewer/save and /vis/viewer/interpolate—see later.



This is a video made with `/control/loop`.

Also uses other `timeWindow` commands:

```
/vis/viewer/set/timeWindow/displayLightFront
 true 0 -500 0 cm -0.5 ns
/vis/viewer/set/timeWindow/displayHeadTime true
/vis/viewer/set/timeWindow/fadeFactor 1
```

See command guidance for detailed descriptions.

Also used: `/vis/modeling/trajectories`

```
/vis/modeling/trajectories/create/drawByParticleID TimeSliceByParticleID
/vis/modeling/trajectories/TimeSliceByParticleID/set e- red
/vis/modeling/trajectories/TimeSliceByParticleID/set gamma green
/vis/modeling/trajectories/TimeSliceByParticleID/set e+ blue
/vis/modeling/trajectories/TimeSliceByParticleID/set pi- cyan
/vis/modeling/trajectories/TimeSliceByParticleID/set pi+ magenta
/vis/modeling/trajectories/TimeSliceByParticleID/set proton firebrick
/vis/modeling/trajectories/TimeSliceByParticleID/set neutron yellow
#/vis/modeling/trajectories/TimeSliceByParticleID/default/setDrawStepPts true
/vis/modeling/trajectories/TimeSliceByParticleID/default/setStepPtsSize 5
#/vis/modeling/trajectories/TimeSliceByParticleID/default/setDrawAuxPts true
/vis/modeling/trajectories/TimeSliceByParticleID/default/setAuxPtsSize 5
/vis/modeling/trajectories/TimeSliceByParticleID/default/setTimeSliceInterval
 0.001 ns
```



# /vis/viewer/save and /vis/viewer/interpolate

- These are old commands (since 10.3, I think) but timeWindow parameters have been brought under their control.
- /vis/viewer/save [<filename> |-]
  - Produces a macro file consisting of Geant4 commands that define the view.
  - This can be read back with /control/execute <filename>
  - Useful for saving view parameters (view direction, zoom, etc.) between runs.
  - Also useful for saving views for /vis/viewer/interpolate
  - The default filenames are g4\_00.g4view, g4\_01.g4view, etc., which can be understood by /vis/viewer/interpolate.



## Example of a save file...

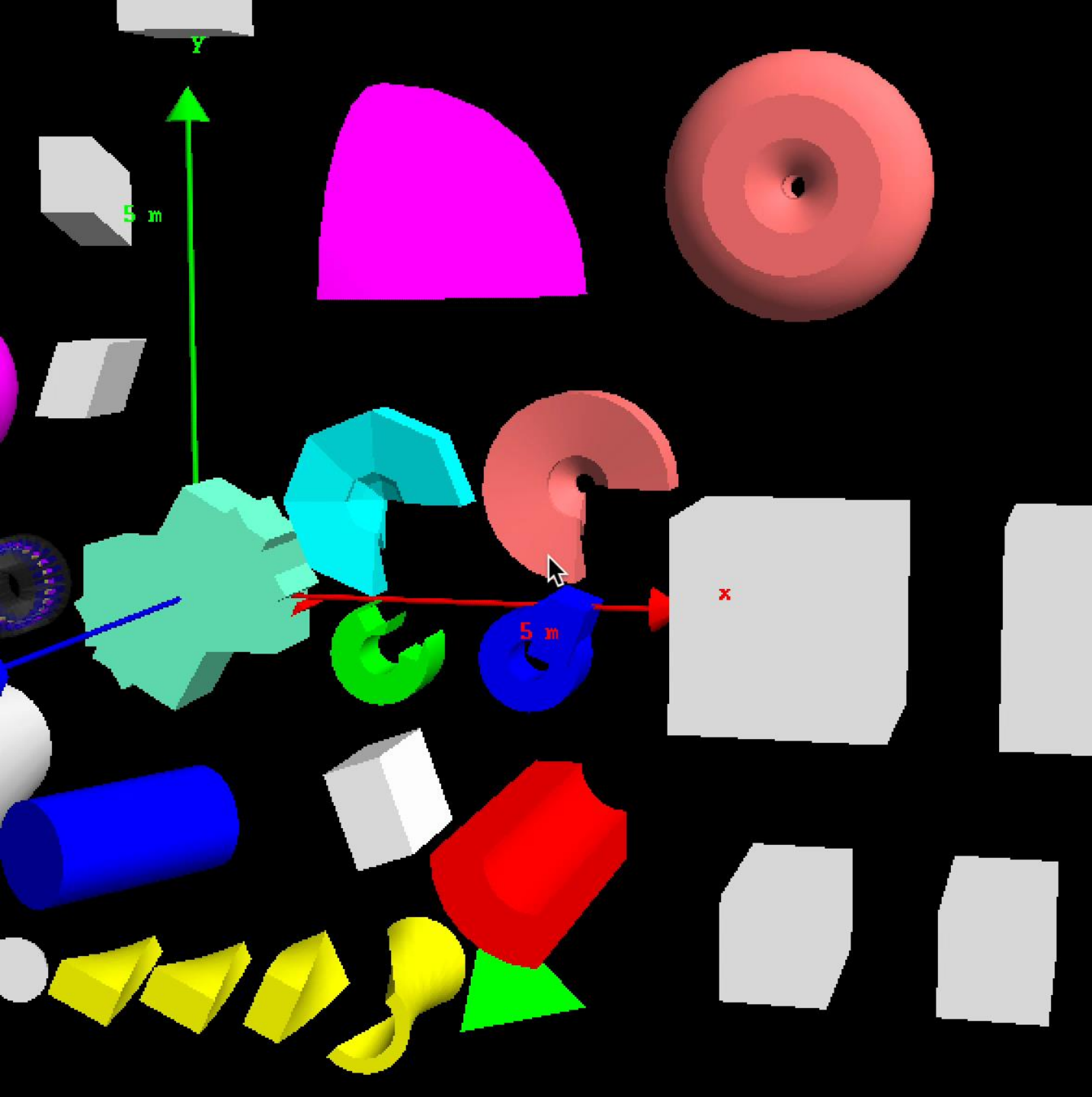
```
#
Camera and lights commands
/vis/viewer/set/viewpointVector 0.527903 0.0871557 0.844821
/vis/viewer/set/upVector 0 1 0
/vis/viewer/set/projection perspective 30 deg
/vis/viewer/zoomTo 30.3989
/vis/viewer/scaleTo 1 1 1
/vis/viewer/set/targetPoint 2.77129 -0.286852 1.00487 m
Note that if you have not set a target point, the vis system sets
a target point based on the scene - plus any panning and dollying -
so don't be alarmed by strange coordinates here.
/vis/viewer/dollyTo 0 fm
/vis/viewer/set/lightsMove object
/vis/viewer/set/lightsVector 1 1 1
/vis/viewer/set/rotationStyle constrainUpDirection
/vis/viewer/set/background 0 0 0 1
/vis/viewer/set/defaultColour 1 1 1 1
/vis/viewer/set/defaultTextColour 0 0 1 1
#
Drawing style commands
/vis/viewer/set/style surface
/vis/viewer/set/hiddenEdge false
/vis/viewer/set/auxilliaryEdge false
/vis/viewer/set/hiddenMarker false
/vis/viewer/set/globalLineWidthScale 1
/vis/viewer/set/globalMarkerScale 1
```

```
#
Scene-modifying commands
/vis/viewer/set/culling global true
/vis/viewer/set/culling invisible true
/vis/viewer/set/culling density false
/vis/viewer/set/culling coveredDaughters false
/vis/viewer/colourByDensity 0 g/cm3
/vis/viewer/set/sectionPlane off
/vis/viewer/set/cutawayMode union
/vis/viewer/clearCutawayPlanes
No cutaway planes defined.
/vis/viewer/set/explodeFactor 1 0 0 0 fm
/vis/viewer/set/lineSegmentsPerCircle 24
#
Touchable commands
/vis/viewer/clearVisAttributesModifiers
/vis/set/touchable expHall_P 0 subtraction_phys 1
/vis/touchable/set/forceWireframe
#
Time window commands
/vis/viewer/set/timeWindow/startTime -1e+100 ns
/vis/viewer/set/timeWindow/endTime 1e+100 ns
/vis/viewer/set/timeWindow/fadeFactor 0
/vis/viewer/set/timeWindow/displayHeadTime false
/vis/viewer/set/timeWindow/displayLightFront false
```

## /vis/viewer/interpolate

- Interpolate views defined by the first argument, which can contain Unix-shell-style pattern matching characters such as '\*', '?' and '|' - see "man sh" and look for "Pattern Matching". The contents of each file are assumed to be "/vis/viewer" commands that specify a particular view. The files are processed in alphanumeric order of filename. The files may be written by hand or produced by the "/vis/viewer/save" command.
- The default is to search the working directory for files with a .g4view extension. Another procedure is to assemble view files in a subdirectory, e.g., "myviews"; then they can be interpolated with "/vis/viewer/interpolate myviews/\*".
- To export interpolated views to file for a future possible movie, write "export" as 5th parameter (OpenGL only).
- Uses a cubic spline interpolation (Catmull-Rom) for continuous view parameters.
- You can interpolate the time window to get a movie of moving particles.
- You can change the visibility of volumes.
- You can even interpolate the colour of a touchable (watch it change smoothly from, say red to blue).

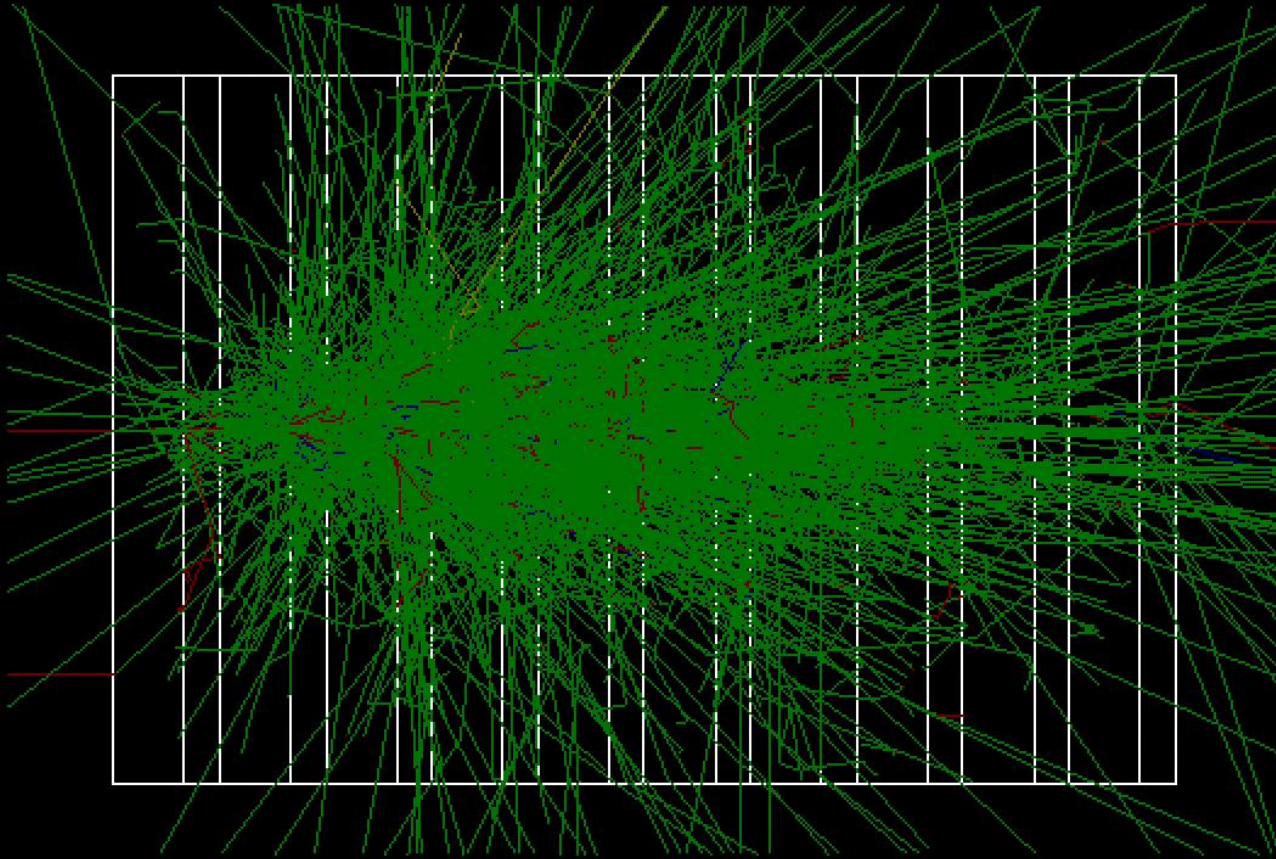




4

## `/vis/viewer/interpolate` – simple example

- With Qt:
  - Choose a view (test202 again)
  - `/vis/viewer/save`
  - Choose another view
  - `/vis/viewer/save`
  - ...and so on...
- Then with Qt or OGLX
  - `/vis/viewer/interpolate` (shows movie on the screen)
- To capture, use QuickTime “New Screen Recording”
  - I had to paste into the session window  
`/control/shell sleep 10`  
`/vis/viewer/interpolate`  
to give me time to hit “Record”.
- Or make lots of files
  - `/vis/viewer/interpolate ! ! ! export`
  - Then with iMovie, “Import media”, select the files in numerical order. Make a project. (Somehow I got iMovie to ascribe a duration of 0.1s (10 fps) to each pdf file. The default seems to be 10s.)
  - Share to file (makes an mp4).
  - Play movie at x2 (20 fps) for smoother motion.



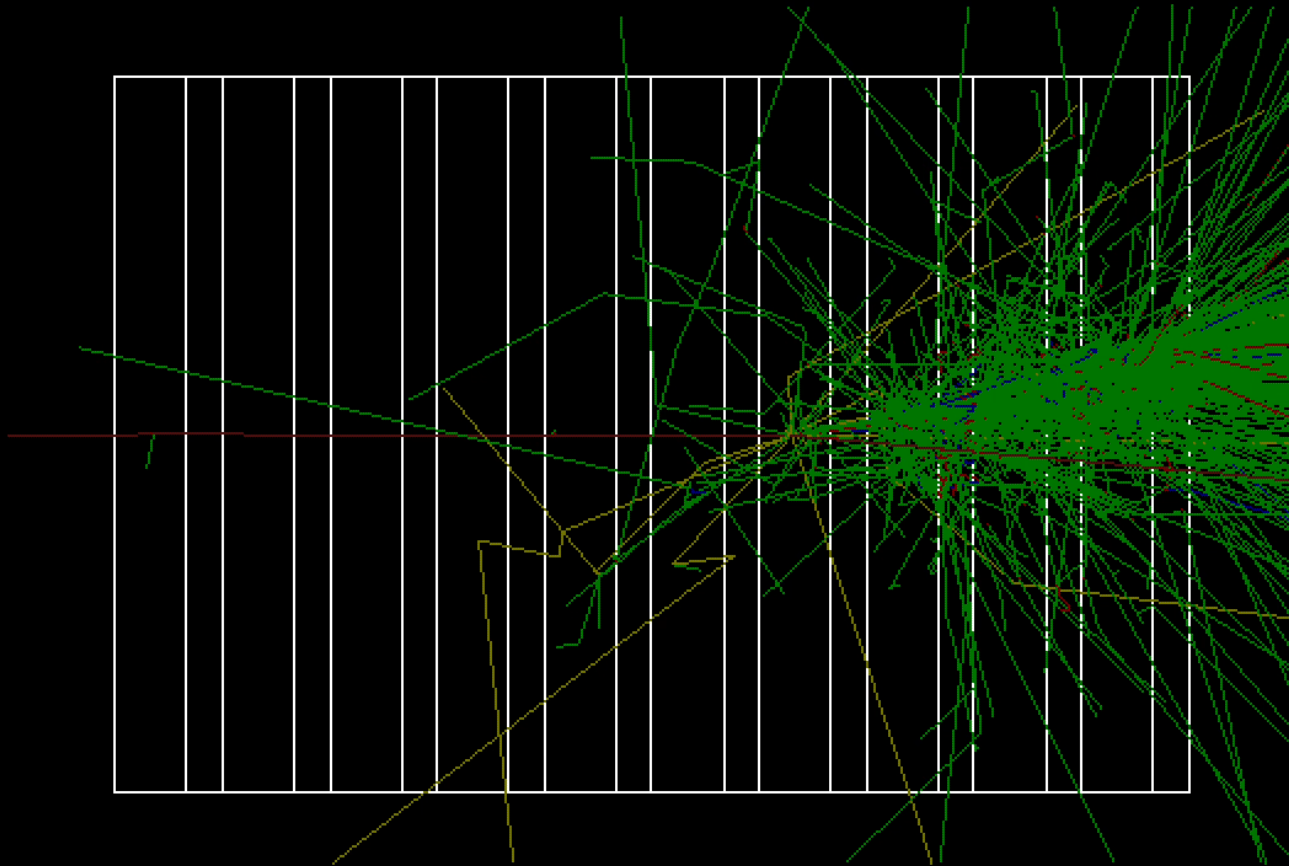
## /vis/viewer/interpolate – time window example

```
/vis/scene/add/trajectories rich
/vis/modeling/trajectories/create/drawByParticleID
TimeSliceByParticleID
/vis/modeling/trajectories/TimeSliceByParticleID/set e- red
/vis/modeling/trajectories/TimeSliceByParticleID/set gamma green
.. # as before
/vis/modeling/trajectories/TimeSliceByParticleID/default/setTimeSliceInterval
 0.001 ns..
/vis/viewer/set/timeWindow/displayLightFront true 0 0 -90 mm 0 ns
/vis/viewer/set/timeWindow/displayHeadTime true
/vis/viewer/set/timeWindow/fadeFactor 1

/gun/energy 10 GeV
/vis/ogl/set/displayListLimit 5000000
/run/beamOn 1

/vis/viewer/set/timeWindow/startTime 0 ns 0.1 ns
/vis/viewer/save
/vis/viewer/set/timeWindow/startTime 0.1 ns 0.1 ns
/vis/viewer/save
..
/vis/viewer/set/timeWindow/startTime 0.9 ns 0.1 ns
/vis/viewer/save

/control/shell sleep 10
/vis/viewer/interpolate
```



# Same as last slide except..  
/gun/energy 10 GeV  
/gun/particle proton

- 10 Gev proton again.
- This time there is some change of view between saves (with Qt). Drawn with OGLX.

```
/vis/viewer/set/timeWindow/startTime 0 ns 0.1 ns
Change view
/vis/viewer/save
/vis/viewer/set/timeWindow/startTime 0.1 ns 0.1 ns
Change view
/vis/viewer/save
..
/vis/viewer/set/timeWindow/startTime 0.9 ns 0.1 ns
Change view
/vis/viewer/save
```

- Have fun