Introduction to Geant4 Visualization and User Interface, part II

I. Hrivnacova, IJCLab Orsay

Most of slides from Laurent GARNIER (IRISA) and Joseph Perl (SLAC)

GEANT4
A SIMULATION TOOLKIT

Outline

- vis.mac commands
  - The /vis/viewer Commands
  - Axes, Trajectories and Hits
  - Visualizing Step Points
  - Smooth trajectories
  - Basic Trajectory Modeling
  - Accumulating Trajectories and Hits
  - Filtering Trajectories
- Compound Commands
- Time slicing
- Interpolation of saved views
- Histograms Plotting
- Extending GUI

vis.mac Macro example

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
#/vis/viewer/set/viewpointThetaPhi 90. 0.
#/vis/viewer/set/style wireframe
#/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
#/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
#/vis/modeling/trajectories/create/drawByParticleID
#/vis/modeling/trajectories/drawByParticleID-0/set e- blue
#/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

vis.mac Macro example (2)

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
#/vis/viewer/set/viewpointThetaPhi 90. 0.
#/vis/viewer/set/style wireframe
#/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
#/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
#/vis/modeling/trajectories/create/drawByParticleID
#/vis/modeling/trajectories/drawByParticleID-0/set e- blue
#/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

What we've covered so far

vis.mac Macro example (3)

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
#/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
#/vis/modeling/trajectories/create/drawByParticleID
#/vis/modeling/trajectories/drawByParticleID-0/set e- blue
#/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

Controlling the viewpoint

The /vis/viewer/… Commands

- Set view angles
  - **/vis/viewer/set/viewpointThetaPhi <theta> <phi>**
  - for example
    - **/vis/viewer/set/viewpointThetaPhi 90. 0.**
- Set direction from target to lights
  - **/vis/viewer/set/lightsVector <x> <y> <z>**
  - for example
    - **/vis/viewer/set/lightsVector -1 0 0**
- Reset viewpoint
  - **/vis/viewer/reset**
- Zoom
  - **/vis/viewer/zoom <scale factor>**
  - for example
    - **/vis/viewer/zoom 2.**
- Set drawing style
  - **/vis/viewer/set/style <style>**
  - for example
    - **/vis/viewer/set/style wireframe**
    - **/vis/viewer/set/style surface**
  - but note that this will not affect volumes that have style explicitly forced by “setForceWireframe” or “setForceSolid” commands in the C++ code

vis.mac Macro example (4)

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
#/vis/modeling/trajectories/create/drawByParticleID
#/vis/modeling/trajectories/drawByParticleID-0/set e- blue
#/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

Add axes and hits

Axes, Trajectories and Hits

- Axes
  - **/vis/scene/add/axes <x_origin> <y_origin> <z_origin> <size> <units>**
  - for example
    - **/vis/scene/add/axes 0 0 0 1 m**

- Trajectories
  - **/vis/scene/add/trajectories**
  - By default, trajectories are redrawn at every event
    - **/run/beamOn 1**

- Hits
  - **/vis/scene/add/hits**
    - Note that not all examples contain code to create hits,
      so in some cases this command will add nothing to the display

vis.mac Macro example (5)

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
#/vis/modeling/trajectories/create/drawByParticleID
#/vis/modeling/trajectories/drawByParticleID-0/set e- blue
#/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

Visualizing step points

Visualizing Step Points

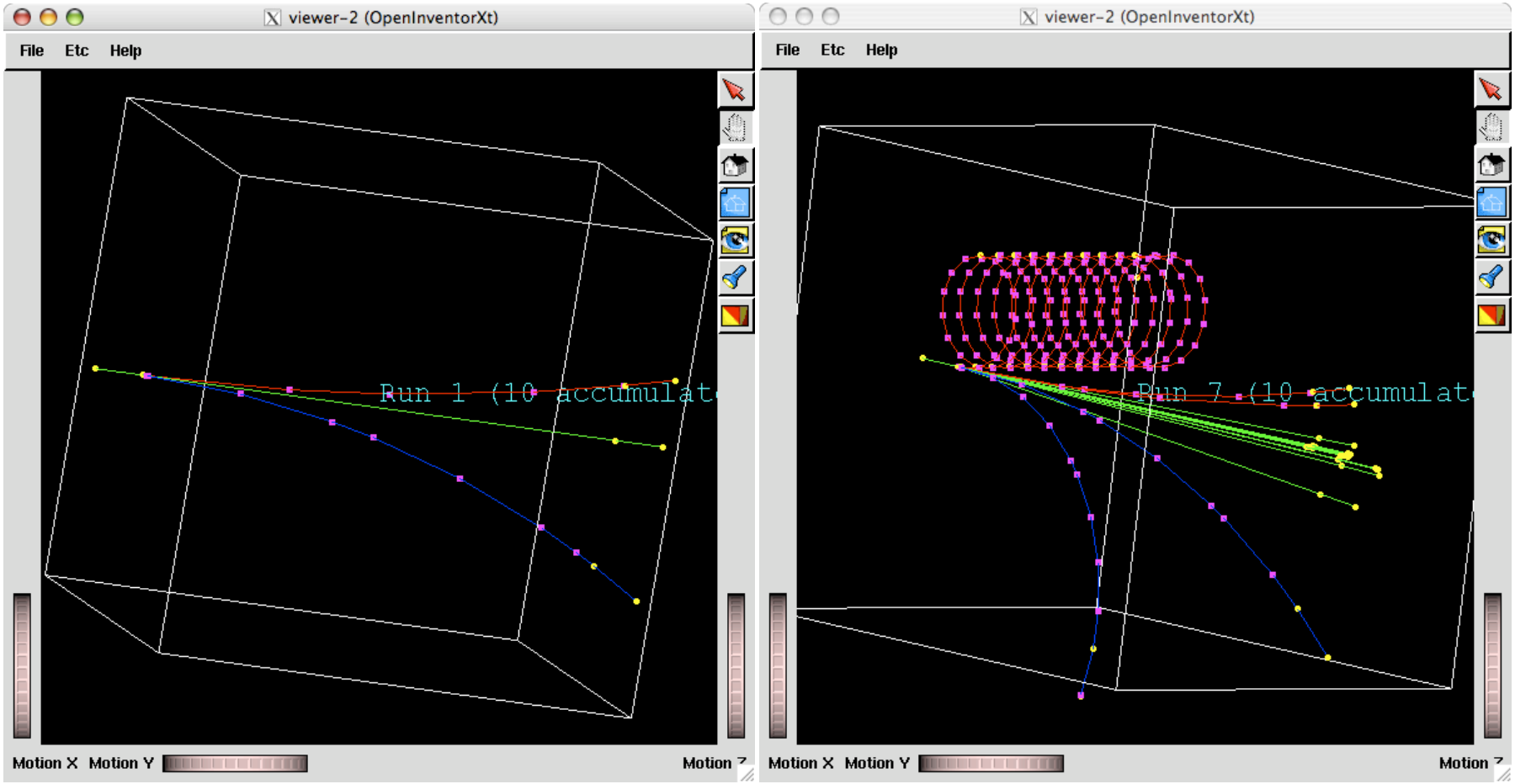
- By default, the trajectory is drawn just as a line
- To also show the step points:
  - **/vis/modeling/trajectories/create/drawByCharge**
  - **/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true**
  - **/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2**
  - This syntax is complicated because it actually supports many more options on how trajectories and step points should be modeled. See Geant4 documentation for more details
- Trajectories and step points can contain additional, non-displayed information
  - such as particle id, momentum, etc.
  - shown when you pick on the trajectory in some visualization drivers.
- This set of information can be made richer by specifying rich trajectories:
  - **/vis/scene/add/trajectories rich**

vis.mac Macro example (6)

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
#/vis/modeling/trajectories/create/drawByParticleID
#/vis/modeling/trajectories/drawByParticleID-0/set e- blue
#/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

Smooth Trajectories

Smooth Trajectory Makes Big Difference for Trajectories that Loop in a Magnetic Field



- Yellow dots are the actual step points used by Geant4
- Magenta dots are auxiliary points added just for purposes of visualization

vis.mac Macro example (7)

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
#/vis/modeling/trajectories/create/drawByParticleID
#/vis/modeling/trajectories/drawByParticleID-0/set e- blue
#/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

Basic trajectory modeling

Basic Trajectory Modeling

- By default, trajectories are color-coded by charge
  - positive = **blue**
  - neutral = **green**
  - negative = **red**
- But you can choose other modeling options, such as color by particle ID
  - **/vis/modeling/trajectories/create/drawByParticleID**
  - **/vis/modeling/trajectories/drawByParticleID-0/set e- blue**
- Right now, when you first turn on drawByParticleID, all particles are set to grey, and you then have to assign any colors you want.

Basic Trajectory Modeling (2)

- Instantiate model
- Configure model
- Register with visualization manager

main.cc

```
// Create and initialise visualization manager
auto visManager = new G4VisExecutive;
visManager->Initialize();


// Create new drawByParticleID model
auto model = new G4TrajectoryDrawByParticleID;

// Configure model
model->SetDefault("cyan");
model->Set("gamma", "green");
model->Set("e+", "magenta");
model->Set("e-", G4Color(0.3, 0.3, 0.3));

//Register model with visualization manager
visManager->RegisterModel(model);
```

vis.mac Macro example (8)

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
#/vis/modeling/trajectories/create/drawByParticleID
#/vis/modeling/trajectories/drawByParticleID-0/set e- blue
#/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

Accumulating trajectories and hits

Accumulating Trajectories and Hits

- By default, you will get a drawing after each event. To instead get just one drawing with all of the accumulated events from that run
  - **/vis/scene/endOfEventAction accumulate**
- This overrides the default and the screen is refreshed before drawing the next event
  - **/vis/scene/endOfEventAction refresh**
- To accumulate the viewer run by run and show drawing at the end of the /run/beamOn, use
  - **/vis/scene/endOfRunAction accumulate**
- Refresh the screen just before drawing the first event of the next run
  - **/vis/scene/endOfRunAction refresh**
- When you actually want to draw, you then have to explicitly issue the command
  - **/vis/viewer/flush**

vis.mac Macro example (9)

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
#/vis/modeling/trajectories/create/drawByParticleID
#/vis/modeling/trajectories/drawByParticleID-0/set e- blue
#/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

Filtering Trajectories

Filtering Trajectories

- By default, all trajectories are drawn
- You apply a filter so that only certain trajectories are drawn:
  - **/vis/filtering/trajectories/create/particleFilter**
  - **/vis/filtering/trajectories/particleFilter-0/add gamma**
- The above adds a filter that only allows gammas to draw
- To instead do the opposite, drawing everything except gammas, include the above, but also add the following:
  - **/vis/filtering/trajectories/particleFilter-0/invert true**

vis.mac Macro example (10)

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
#/vis/modeling/trajectories/create/drawByParticleID
#/vis/modeling/trajectories/drawByParticleID-0/set e- blue
#/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

To avoid excessive redrawing on immediate viewers

reset to «true»

To force output of a new file

vis.mac Macro example (11)

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
#/vis/modeling/trajectories/create/drawByParticleID
#/vis/modeling/trajectories/drawByParticleID-0/set e- blue
#/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

To turn off unwanted visualization messages on the console

To Turn off Unwanted Visualization Messages

- You can control how many messages visualization puts on the console by:
  - **/vis/verbose <level>**
  - **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.**

Complete List of Commands

- This presentation has shown only a very small subset of Geant4 vis commands.
  Even for those commands shown, only a few of the options have been presented.

- Each visualization driver may have its own set of additional commands.
  - Idle> help

- To see the complete set of commands, use the interactive command guidance (i.e., type help and then type the appropriate number for “vis”).

- Note that many of the command details are only loaded into the help system once you start using the given command
  - e.g., when you first look at the help for /vis/modeling, you will see only
    - **/vis/modeling/trajectories/create**
    - **/vis/modeling/trajectories/list**
  - But once you have done your first
    - **/vis/modeling/trajectories/create/drawByParticleID**
  - you will see many subcommands such as
    - **/vis/modeling/trajectories/drawByParticleID-0/set**
    - **/vis/modeling/trajectories/drawByParticleID-0/setRGBA**
    - etc.

Compound Commands

- To allow you to work quickly, Geant4 visualization lets you issue the equivalent of several common commands at one time by using a “compound command”.

- Some of the commands you have already seen in this presentation are actually compound commands:
  - **/vis/open**
    - **/vis/sceneHandler/create**
    - **/vis/viewer/create**
  - **/vis/drawVolume**
    - **/vis/scene/create**
    - **/vis/scene/add/volume**
  - **/vis/viewer/flush**
    - **/vis/viewer/refresh**
    - **/vis/viewer/update**

- We mention this just so that you can understand other people’s examples you see that may not contain the familiar /vis/open or /vis/drawVolume

Time slicing

- **movies** example in **extended/visualization**
  - illustrates how to create a movie
  - new in Geant4 11.0, before in basic B4

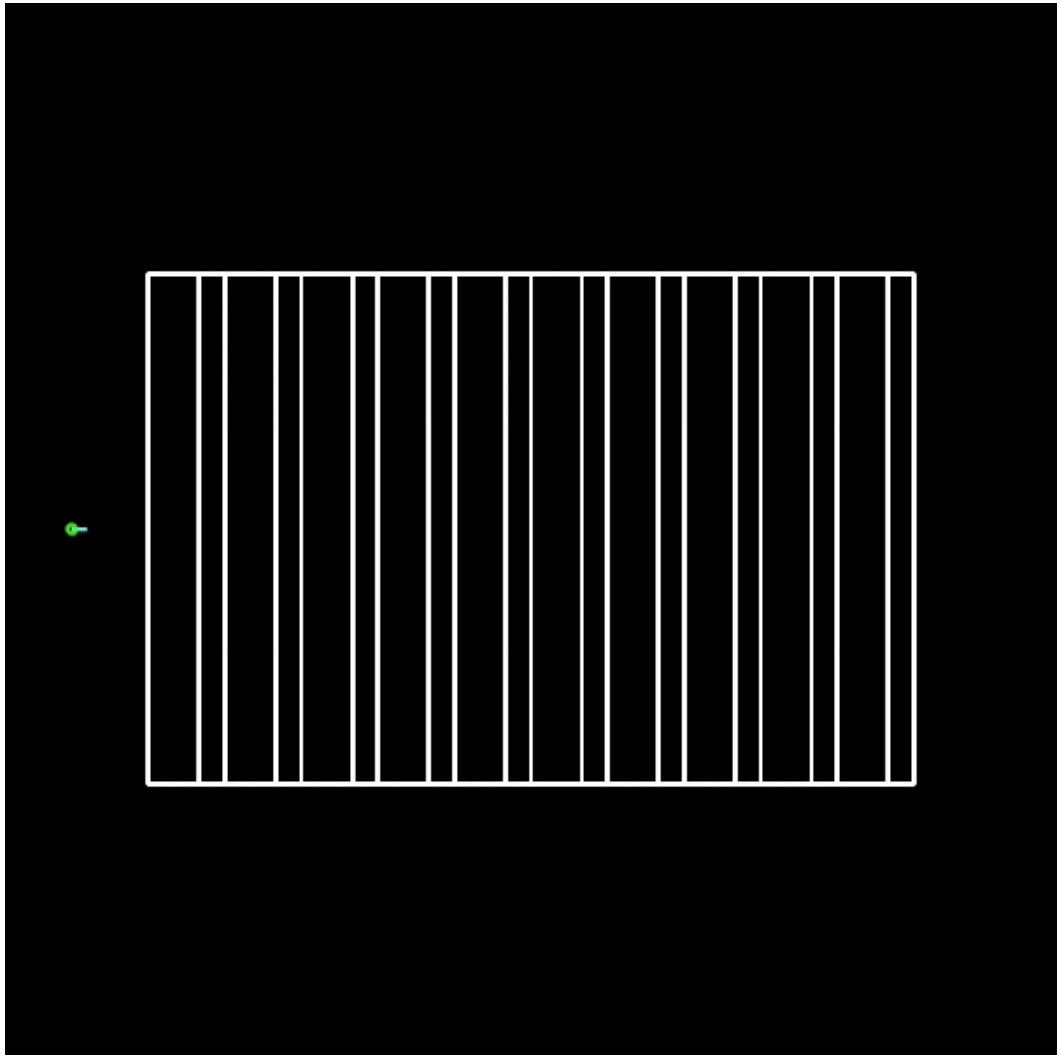
- **visTutor/exN03Vis12.mac**
  a) Draw by charge with trajectory points
  b) Draw by particle ID (remove γ's)
  c) π-μ-e decay

- **visTutor/exN03Vis13.mac**
  – 10 GeV EM shower showing light front
  – Camera follows (pans) at speed of light

- *To run these macros the visTutor directory has to be first copied in the example build area and then the macro should be run in batch mode:*

```
./movies -m visTutor/exN03Vis12[13].mac
```

Time slicing (2)



Hadronic physics

Interacts early (potential confusion with EM shower)

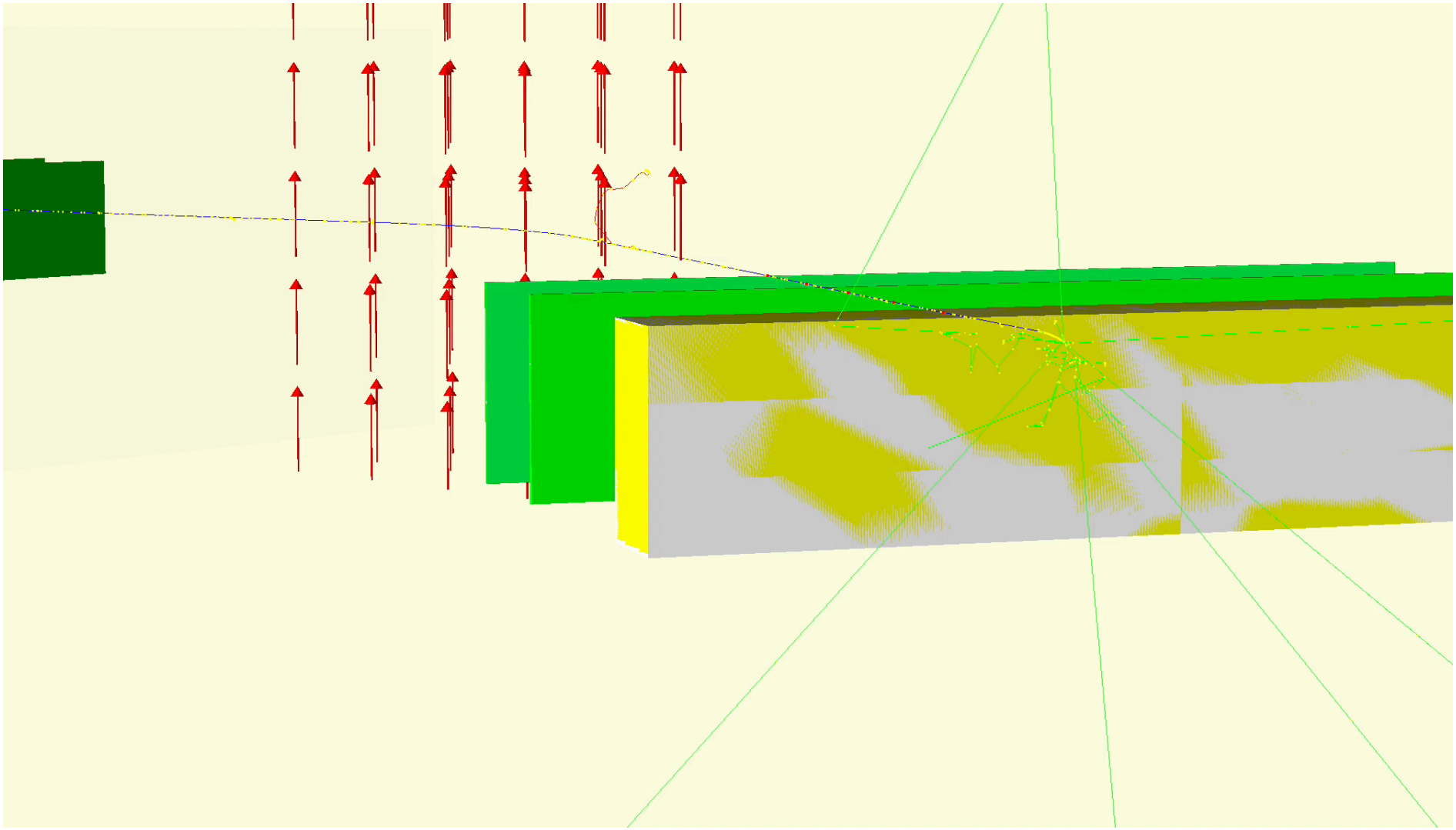
Produces EM shower, presumably via charge exchange to $\pi^0 \rightarrow \gamma\gamma$

Neutrons also produced

π^+ magenta
π^- cyan
n yellow
ν green
Others grey
Duration 2 ns

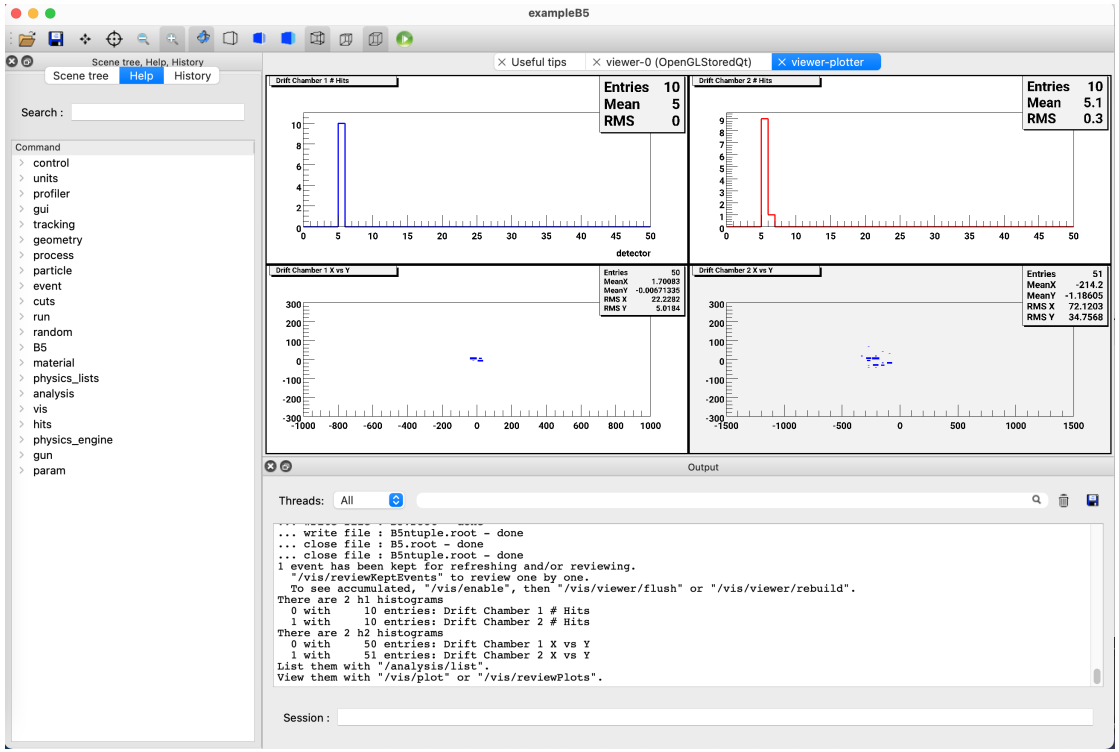
Interpolation of saved views

- **Save a sequence of views**
  - with or without events (trajectories)
  - for each view:
    /vis/viewer/save
  - view parameters are saved to a sequence of files
    - g4_00.view, g4_01.view, etc.
- **/vis/viewer/interpolate**
  - with or without the same or different events
- **/vis/viewer/interpolate ! ! ! ! export**
  - produces G4OpenGL_viewer-0_nnnn.pdf (default 50 per saved view)
  - make a movie with iMovie, for example
    - set “duration” of each file to 0.1 s (this seems to be the minimum)
    - play it back at ×2 or ×4



Histograms Plotting

- Since version 11, the Geant4 visualization system is equipped to be able to do plotting, then to have a representation (a plot) of 1D or 2D histograms within a Geant4 visualization viewer.
- Currently only new **ToolsSG** visualization driver has this feature
- Demonstrated in the basic B5 example **`plotter.mac`** macro

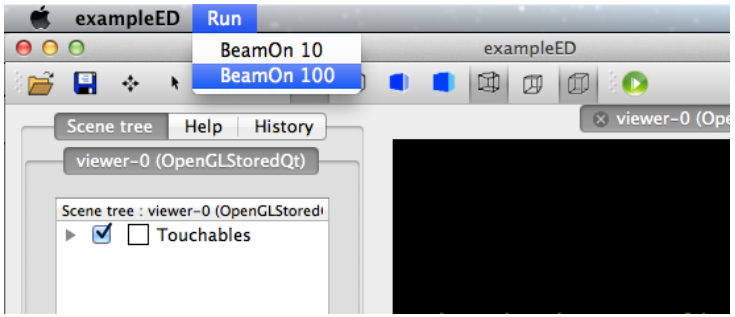


plotter.mac Macro example

```
# activate the TSG vis driver
/vis/sceneHandler/create TSG scene-handler-plotter

# create the viewer
/vis/viewer/create scene-handler-plotter viewer-plotter 600x600-0+0
/vis/viewer/set/background 1 1 1
/vis/viewer/zoomTo 1
/vis/viewer/set/viewpointVector 0 0 1

# create the scene and add the plotter
/vis/plotter/create plotter-0
/vis/scene/create scene-plotter
/vis/scene/add/plotter plotter-0
/vis/sceneHandler/attach scene-plotter

# create plotter regions and attach histograms to them
/vis/plotter/setLayout plotter-0 2 2  # create 2x2 plotting regions.
/vis/plotter/add/h1 0 plotter-0 0  # add h1 with id=0
/vis/plotter/add/h1 1 plotter-0 1  # add h1 with id=1
/vis/plotter/add/h2 0 plotter-0 2  # add h2 with id=0
/vis/plotter/add/h2 1 plotter-0 3  # add h2 with id=1

# plotters customisation (and many more, see B5/plotter.mac)
/vis/plotter/style/list
/vis/plotter/addStyle plotter-0 reset
/vis/plotter/addStyle plotter-0 ROOT_default
```

Extend your GUI

- Add menus in your graphical user interface



/gui/addMenu Run Run
/gui/addButton Run «BeamOn 10» «/run/beamOn 10»
/gui/addButton Run «BeamOn 100» «/run/beamOn 100»

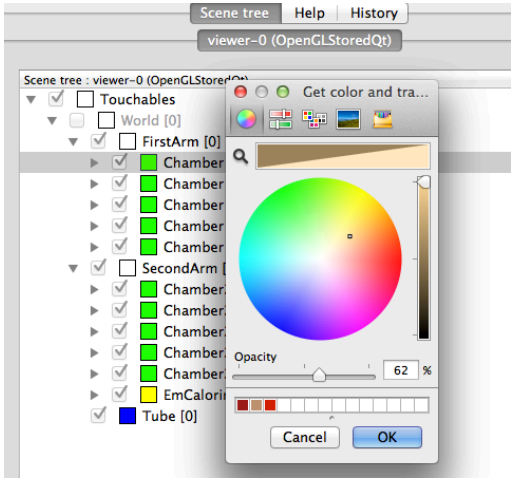
- Usually, commands setting GUI are grouped in a macro file called «gui.mac»

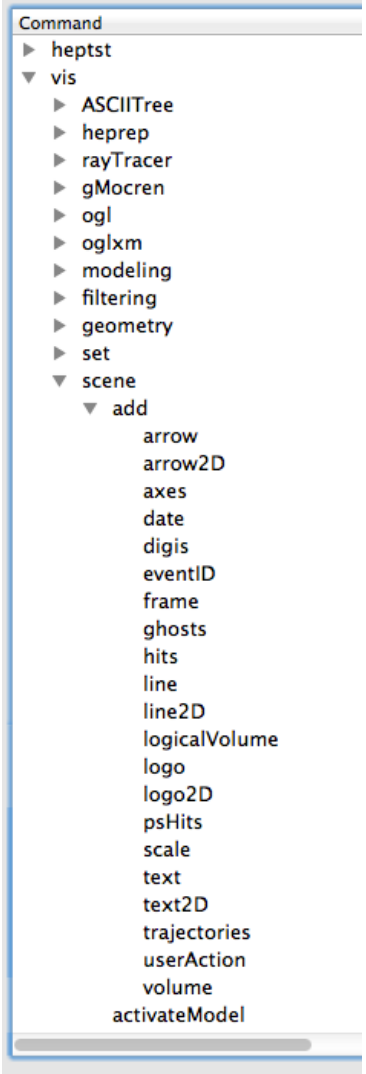
- Add icons in your GUI (available only for Qt driver):
  - **/gui/addIcon**
    - predefine icons: move, pick, zoom_out, zoom_in, rotate, hidden_line_removal, hidden_line_and_surface_removal, solid, wireframe, perspective, ortho
- User defined icons :
  - **/gui/addIcon "Run beam on" user_icon "/run/beamOn 1" run.png**

Extend your GUI (2)

- Add extras :
  - **Axes, Date, Scale, Text, Text2D...**
- Export viewer in vectorial/non vectorial format (only for OpenGL Viewers)
  - Change size up to 8192*8192 (your max OpenGL card capacity)
  - **/vis/ogl/export <name.format> <width> <height>**
  - Available formats : eps/pdf/svg/ps, jpg, jp2, png, …
- Change color on volumes
  - By GUI
  - By commands :

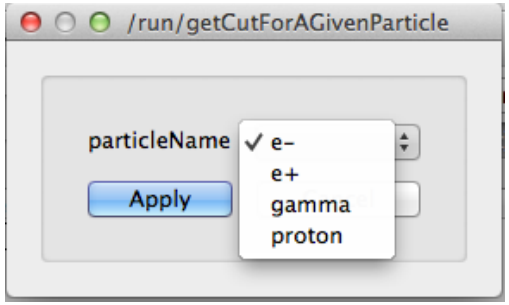
/vis/geometry/set/colour logical-volume-name depth red green blue opacity





Extend your GUI, Qt special

- /gui/addButton or /gui/addMenu without any parameters will open a popup to choose parameters
  - `/gui/addButton Run "Set cuts for a given particle" "/run/getCutForAGivenParticle"`



  - `/addButton Run "change background color" "/vis/viewer/set/background"`