

Introduction to Geant4 Visualization and User Interface, part II

(based on Geant4.10.4)

Laurent GARNIER,

IRISA / INS2I / CNRS

Based on Joseph Perl (SLAC) slides



GEANT4
A SIMULATION TOOLKIT

Launching sequence

Geant4 version Name: geant4-10-04-ref-00 [MT] (08-December-2017)

<< in Multi-threaded mode >>

Copyright : Geant4 Collaboration

Reference : NIM A 506 (2003), 250-303

WWW : <http://cern.ch/geant4>

<<< Geant4 Physics List simulation engine: FTFP_BERT 2.0

Using Root

Visualization Manager instantiating with verbosity "warnings (3)"...

Visualization Manager initialising...

Registering graphics systems...

You have successfully registered the following graphics systems.

Current available graphics systems are:

ASCIITree (ATree)

DAWNFILE (DAWNFILE)

G4HepRep (HepRepXML)

G4HepRepFile (HepRepFile)

RayTracer (RayTracer)

VRML1FILE (VRML1FILE)

VRML2FILE (VRML2FILE)

gMocrenFile (gMocrenFile)

OpenGLImmediateQt (OGLIQt, OGLI)

OpenGLStoredQt (OGLSQt, OGL, OGLS)

Registering model factories...

You have successfully registered the following model factories.

Registered model factories:

generic

drawByAttribute

drawByCharge

drawByOriginVolume

drawByParticleID

drawByEncounteredVolume

Registered filter factories:

attributeFilter

chargeFilter

originVolumeFilter

particleFilter

encounteredVolumeFilter

You have successfully registered the following user vis actions.

Run Duration User Vis Actions: none

End of Event User Vis Actions: none

End of Run User Vis Actions: none

Some /vis commands (optionally) take a string to specify colour.

"/vis/list" to see available colours.

Launch your application

Geant4 version

PhysList simulation

Register graphic systems

Register model factories

Register filter factories

Register user vis actions

Launching sequence

Geant4 version Name: geant4-10-04-ref-00 [MT] (08-December-2017)

<< in Multi-threaded mode >>

Copyright : Geant4 Collaboration

Reference : NIM A 506 (2003), 250-303

WWW : <http://cern.ch/geant4>

<<< **Geant4 Physics List simulation engine: FTFP_BERT 2.0**

Using Root

Visualization Manager instantiating with verbosity "warnings (3)"...

Visualization Manager initialising...

Registering graphics systems...

You have successfully registered the following graphics systems.

Current available graphics systems are:

ASCIITree (ATree)

DAWNFILE (DAWNFILE)

G4HepRep (HepRepXML)

G4HepRepFile (HepRepFile)

RayTracer (RayTracer)

VRML1FILE (VRML1FILE)

VRML2FILE (VRML2FILE)

gMocrenFile (gMocrenFile)

OpenGLImmediateQt (OGLIQt, OGLI)

OpenGLStoredQt (OGLSQt, OGL, OGLS)

Registering model factories...

You have successfully registered the following model factories.

Registered model factories:

generic

drawByAttribute

drawByCharge

drawByOriginVolume

drawByParticleID

drawByEncounteredVolume

Registered filter factories:

attributeFilter

chargeFilter

originVolumeFilter

particleFilter

encounteredVolumeFilter

You have successfully registered the following user vis actions.

Run Duration User Vis Actions: none

End of Event User Vis Actions: none

End of Run User Vis Actions: none

Some /vis commands (optionally) take a string to specify colour.

"/vis/list" to see available colours.

Launch your application

Geant4 version

PhysList simulation

Register graphic systems

Register model factories

Register filter factories

Register user vis actions

Launching sequence

Geant4 version Name: geant4-10-04-ref-00 [MT] (08-December-2017)

<< in Multi-threaded mode >>

Copyright : Geant4 Collaboration

Reference : NIM A 506 (2003), 250-303

WWW : <http://cern.ch/geant4>

<<< Geant4 Physics List simulation engine: FTFP_BERT 2.0

Using Root

Visualization Manager instantiating with verbosity "warnings (3)"...

Visualization Manager initialising...

Registering graphics systems...

You have successfully registered the following graphics systems.

Current available graphics systems are:

ASCIITree (ATree)

DAWNFILE (DAWNFILE)

G4HepRep (HepRepXML)

G4HepRepFile (HepRepFile)

RayTracer (RayTracer)

VRML1FILE (VRML1FILE)

VRML2FILE (VRML2FILE)

gMocrenFile (gMocrenFile)

OpenGLImmediateQt (OGLIQt, OGLI)

OpenGLStoredQt (OGLSQt, OGL, OGLS)

Registering model factories...

You have successfully registered the following model factories.

Registered model factories:

generic

drawByAttribute

drawByCharge

drawByOriginVolume

drawByParticleID

drawByEncounteredVolume

Registered filter factories:

attributeFilter

chargeFilter

originVolumeFilter

particleFilter

encounteredVolumeFilter

You have successfully registered the following user vis actions.

Run Duration User Vis Actions: none

End of Event User Vis Actions: none

End of Run User Vis Actions: none

Some /vis commands (optionally) take a string to specify colour.

"/vis/list" to see available colours.

Launch your application

Geant4 version

PhysList simulation

Register graphic systems

Register model factories

Register filter factories

Register user vis actions

Launching sequence

Geant4 version Name: geant4-10-04-ref-00 [MT] (08-December-2017)

<< in Multi-threaded mode >>

Copyright : Geant4 Collaboration

Reference : NIM A 506 (2003), 250-303

WWW : <http://cern.ch/geant4>

<<< Geant4 Physics List simulation engine: FTFP_BERT 2.0

Using Root

Visualization Manager instantiating with verbosity "warnings (3)"...

Visualization Manager initialising...

Registering graphics systems...

You have successfully registered the following graphics systems.

Current available graphics systems are:

ASCIITree (ATree)

DAWNFILE (DAWNFILE)

G4HepRep (HepRepXML)

G4HepRepFile (HepRepFile)

RayTracer (RayTracer)

VRML1FILE (VRML1FILE)

VRML2FILE (VRML2FILE)

gMocrenFile (gMocrenFile)

OpenGLImmediateQt (OGLIQt, OGLI)

OpenGLStoredQt (OGLSQt, OGL, OGLS)

Registering model factories...

You have successfully registered the following model factories.

Registered model factories:

generic

drawByAttribute

drawByCharge

drawByOriginVolume

drawByParticleID

drawByEncounteredVolume

Registered filter factories:

attributeFilter

chargeFilter

originVolumeFilter

particleFilter

encounteredVolumeFilter

You have successfully registered the following user vis actions.

Run Duration User Vis Actions: none

End of Event User Vis Actions: none

End of Run User Vis Actions: none

Some /vis commands (optionally) take a string to specify colour.

"/vis/list" to see available colours.

Launch your application

Geant4 version

PhysList simulation

Register graphic systems

Register model factories

Register filter factories

Register user vis actions

Launching sequence

Geant4 version Name: geant4-10-04-ref-00 [MT] (08-December-2017)

<< in Multi-threaded mode >>

Copyright : Geant4 Collaboration

Reference : NIM A 506 (2003), 250-303

WWW : <http://cern.ch/geant4>

<<< Geant4 Physics List simulation engine: FTFP_BERT 2.0

Using Root

Visualization Manager instantiating with verbosity "warnings (3)"...

Visualization Manager initialising...

Registering graphics systems...

You have successfully registered the following graphics systems.

Current available graphics systems are:

ASCIITree (ATree)

DAWNFILE (DAWNFILE)

G4HepRep (HepRepXML)

G4HepRepFile (HepRepFile)

RayTracer (RayTracer)

VRML1FILE (VRML1FILE)

VRML2FILE (VRML2FILE)

gMocrenFile (gMocrenFile)

OpenGLImmediateQt (OGLIQt, OGLI)

OpenGLStoredQt (OGLSQt, OGL, OGLS)

Registering model factories...

You have successfully registered the following model factories.

Registered model factories:

generic

drawByAttribute

drawByCharge

drawByOriginVolume

drawByParticleID

drawByEncounteredVolume

Registered filter factories:

attributeFilter

chargeFilter

originVolumeFilter

particleFilter

encounteredVolumeFilter

You have successfully registered the following user vis actions.

Run Duration User Vis Actions: none

End of Event User Vis Actions: none

End of Run User Vis Actions: none

Some /vis commands (optionally) take a string to specify colour.

"/vis/list" to see available colours.

Launch your application

Geant4 version

PhysList simulation

Register graphic systems

Register model factories

Register filter factories

Register user vis actions

Launching sequence

Geant4 version Name: geant4-10-04-ref-00 [MT] (08-December-2017)

<< in Multi-threaded mode >>

Copyright : Geant4 Collaboration

Reference : NIM A 506 (2003), 250-303

WWW : <http://cern.ch/geant4>

<<< Geant4 Physics List simulation engine: FTFP_BERT 2.0

Using Root

Visualization Manager instantiating with verbosity "warnings (3)"...

Visualization Manager initialising...

Registering graphics systems...

You have successfully registered the following graphics systems.

Current available graphics systems are:

ASCIITree (ATree)

DAWNFILE (DAWNFILE)

G4HepRep (HepRepXML)

G4HepRepFile (HepRepFile)

RayTracer (RayTracer)

VRML1FILE (VRML1FILE)

VRML2FILE (VRML2FILE)

gMocrenFile (gMocrenFile)

OpenGLImmediateQt (OGLIQt, OGLI)

OpenGLStoredQt (OGLSQt, OGL, OGLS)

Registering model factories...

You have successfully registered the following model factories.

Registered model factories:

generic

drawByAttribute

drawByCharge

drawByOriginVolume

drawByParticleID

drawByEncounteredVolume

Registered filter factories:

attributeFilter

chargeFilter

originVolumeFilter

particleFilter

encounteredVolumeFilter

You have successfully registered the following user vis actions.

Run Duration User Vis Actions: none

End of Event User Vis Actions: none

End of Run User Vis Actions: none

Some /vis commands (optionally) take a string to specify colour.

"/vis/list" to see available colours.

Launch your application

Geant4 version

PhysList simulation

Register graphic systems

Register model factories

Register filter factories

Register user vis actions

Launching sequence

Kernel initialization
Call Construct()

Overlaps checking (if set)

```
/control/saveHistory
/run/verbose 2
Visualization verbosity changed to errors (2)
#
# Change the default number of threads (in multi-threaded mode)
#/run/numberOfThreads 4
#
# Initialize kernel
/run/initialize
userDetector->Construct() start.
***** Table : Nb of materials = 3 *****
Material: G4_Pb density: 11.350 g/cm3 RadL: 5.613 mm Nucl.Int.Length: 18.248 cm
          lmean: 823.000 eV temperature: 293.15 K pressure: 1.00 atm

---> Element: Pb (Pb) Z = 82.0 N = 207 A = 207.217 g/mole
---> Isotope: Pb204 Z = 82 N = 204 A = 203.97 g/mole abundance: 1.400 %
---> Isotope: Pb206 Z = 82 N = 206 A = 205.97 g/mole abundance: 24.100 %
---> Isotope: Pb207 Z = 82 N = 207 A = 206.98 g/mole abundance: 22.100 %
---> Isotope: Pb208 Z = 82 N = 208 A = 207.98 g/mole abundance: 52.400 %
      ElmMassFraction: 100.00 % ElmAbundance 100.00 %
Material: liquidArgon density: 1.390 g/cm3 RadL: 14.065 cm Nucl.Int.Length: 86.078 cm
          lmean: 188.000 eV temperature: 293.15 K pressure: 1.00 atm

---> Element: Ar (Ar) Z = 18.0 N = 40 A = 39.950 g/mole
---> Isotope: Ar36 Z = 18 N = 36 A = 35.97 g/mole abundance: 0.337 %
---> Isotope: Ar38 Z = 18 N = 38 A = 37.96 g/mole abundance: 0.063 %
---> Isotope: Ar40 Z = 18 N = 40 A = 39.96 g/mole abundance: 99.600 %
      ElmMassFraction: 100.00 % ElmAbundance 100.00 %
Material: Galactic density: 0.000 kg/m3 RadL: 204727512.315 pc Nucl.Int.Length: 113427275.267 pc
          lmean: 19.200 eV temperature: 2.73 K pressure: 0.00 atm

---> Element: H (H) Z = 1.0 N = 1 A = 1.010 g/mole
---> Isotope: H1 Z = 1 N = 1 A = 1.01 g/mole abundance: 99.989 %
---> Isotope: H2 Z = 1 N = 2 A = 2.01 g/mole abundance: 0.011 %
      ElmMassFraction: 100.00 % ElmAbundance 100.00 %
Checking overlaps for volume Calorimeter ... OK!
Checking overlaps for volume Abso ... OK!
Checking overlaps for volume Gap ... OK!
```


Launching sequence

Kernel initialization
Call Construct()

Overlaps checking (if set)

```
/control/saveHistory
/run/verbose 2
Visualization verbosity changed to errors (2)
#
# Change the default number of threads (in multi-threaded mode)
#/run/numberOfThreads 4
#
# Initialize kernel
/run/initialize
userDetector->Construct() start.
***** Table : Nb of materials = 3 *****
Material:  G4_Pb   density: 11.350 g/cm3  RadL: 5.613 mm  Nucl.Int.Length: 18.248 cm
            lmean: 823.000 eV  temperature: 293.15 K  pressure: 1.00 atm

---> Element: Pb (Pb)  Z = 82.0  N = 207  A = 207.217 g/mole
---> Isotope: Pb204  Z = 82  N = 204  A = 203.97 g/mole  abundance: 1.400 %
---> Isotope: Pb206  Z = 82  N = 206  A = 205.97 g/mole  abundance: 24.100 %
---> Isotope: Pb207  Z = 82  N = 207  A = 206.98 g/mole  abundance: 22.100 %
---> Isotope: Pb208  Z = 82  N = 208  A = 207.98 g/mole  abundance: 52.400 %
      ElmMassFraction: 100.00 %  ElmAbundance 100.00 %
Material: liquidArgon  density: 1.390 g/cm3  RadL: 14.065 cm  Nucl.Int.Length: 86.078 cm
            lmean: 188.000 eV  temperature: 293.15 K  pressure: 1.00 atm

---> Element: Ar (Ar)  Z = 18.0  N = 40  A = 39.950 g/mole
---> Isotope: Ar36  Z = 18  N = 36  A = 35.97 g/mole  abundance: 0.337 %
---> Isotope: Ar38  Z = 18  N = 38  A = 37.96 g/mole  abundance: 0.063 %
---> Isotope: Ar40  Z = 18  N = 40  A = 39.96 g/mole  abundance: 99.600 %
      ElmMassFraction: 100.00 %  ElmAbundance 100.00 %
Material: Galactic  density: 0.000 kg/m3  RadL: 204727512.315 pc  Nucl.Int.Length: 113427275.267 pc
            lmean: 19.200 eV  temperature: 2.73 K  pressure: 0.00 atm

---> Element: H (H)  Z = 1.0  N = 1  A = 1.010 g/mole
---> Isotope: H1  Z = 1  N = 1  A = 1.01 g/mole  abundance: 99.989 %
---> Isotope: H2  Z = 1  N = 2  A = 2.01 g/mole  abundance: 0.011 %
      ElmMassFraction: 100.00 %  ElmAbundance 100.00 %
Checking overlaps for volume Calorimeter ... OK!
Checking overlaps for volume Abso ... OK!
Checking overlaps for volume Gap ... OK!
```

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
```

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

What we've covered so far

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
```

Controlling the viewpoint



The `/vis/viewer/...` Commands

- Set view angles
 - `/vis/viewer/set/viewpointThetaPhi <theta_angle> <phi_angle>`
 - for example
 - `/vis/viewer/set/viewpointThetaPhi 90. 0.`
- Zoom
 - `/vis/viewer/zoom <scale factor>`
 - for example
 - `/vis/viewer/zoom 2.`
- Reset viewpoint
 - `/vis/viewer/reset`
- 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/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/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

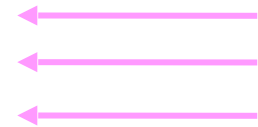
#/vis/scene/endOfEventAction accumulate

/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/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

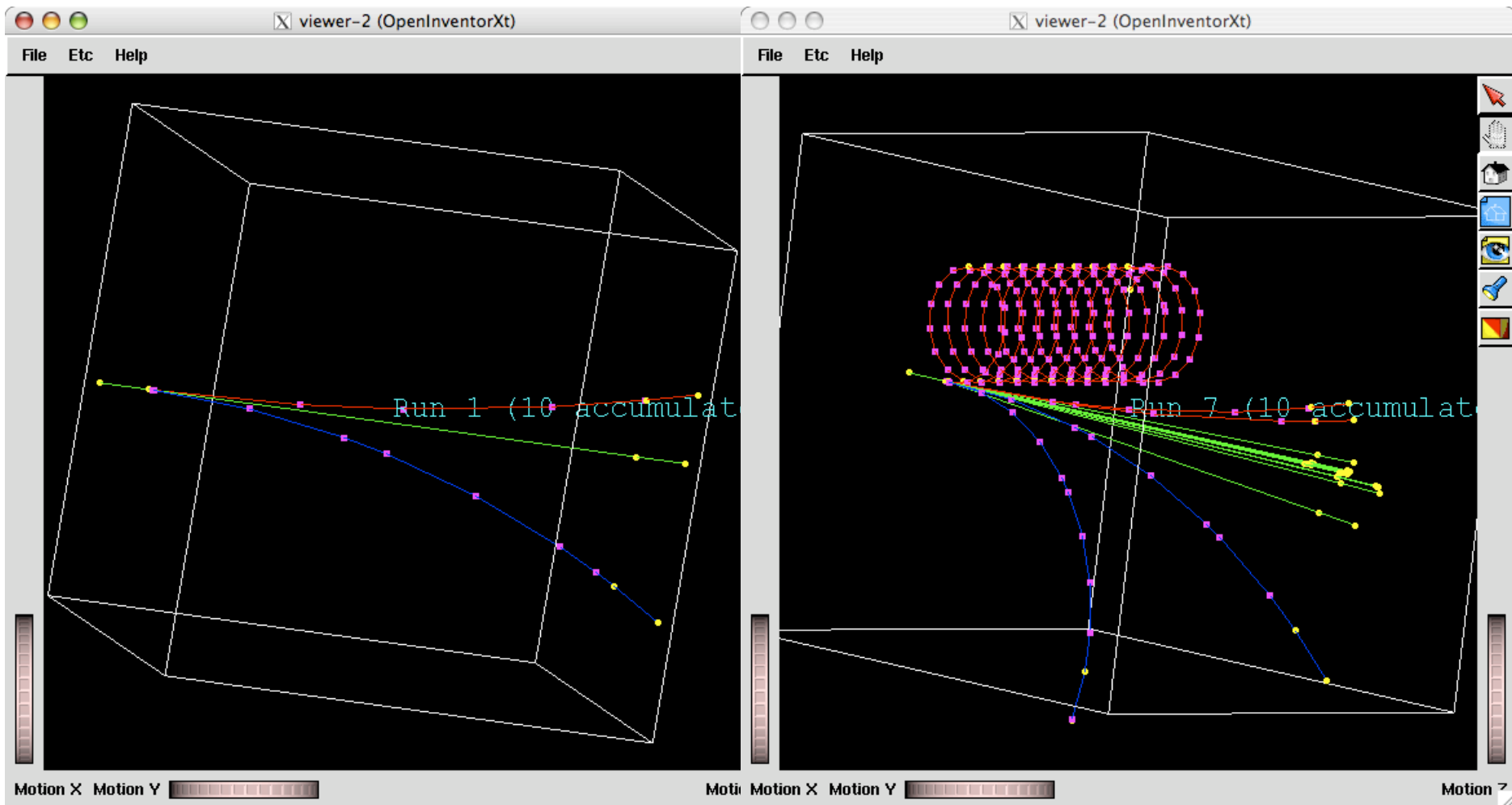
/vis/verbose warnings

#/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/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

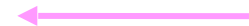
#/vis/scene/endOfEventAction accumulate

/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.

main.cc

```
// Create and initialise visualization manager
G4VisManager* visManager = new G4VisExecutive;
visManager->Initialize();
```

- **Instantiate model**
- **Configure model**
- **Register with visualization manager**

```
// Create new drawByParticleID model
G4TrajectoryDrawByParticleID* 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/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

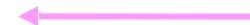
#/vis/scene/endOfEventAction accumulate

/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
 - **/vis/scene/endOfEventAction refresh**
- To even suppress that one drawing from the end of the /run/beamOn, use
 - **/vis/scene/endOfRunAction accumulate**
- This overrides the default
 - **/vis/scene/endOfRunAction refresh**
- When you actually want to draw, you then have to explicitly issue the command
 - **/vis/viewer/flush**

/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

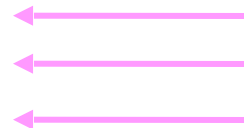
#/vis/scene/endOfEventAction accumulate

/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`
- We'll cover more details on this in a later presentation (including how to filter hits)

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

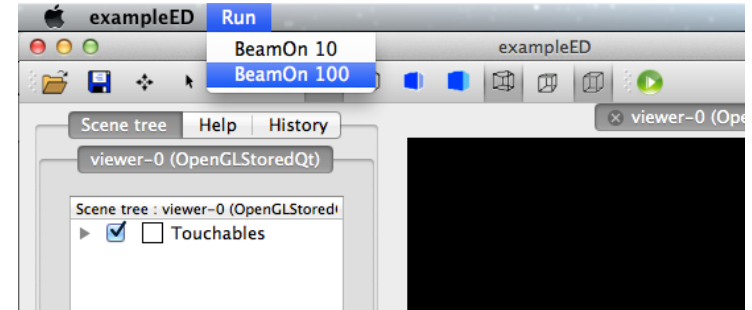
Command Guidance

- Complete guidance on all commands is available from the command line:
 - Idle> help
 - Command directory path : /
 - Sub-directories :
 - 1) /control/ UI control commands.
 - 2) /units/ Available units.
 - 3) /geometry/ Geometry control commands.
 - 4) /tracking/ TrackingManager and SteppingManager control commands.
 - 5) /event/ EventManager control commands.
 - 6) /run/ Run control commands.
 - 7) /random/ Random number status control commands.
 - 8) /particle/ Particle control commands.
 - 9) /process/ Process Table control commands.
 - 10) /vis/ Visualization commands.
 - 11) /mydet/ A01 detector setup control commands.
 - 12) /hits/ Sensitive detectors and Hits
 - 13) /gun/ Particle Gun control commands.
 - Commands :
 - Type the number (0:end, -n:n level back) :
- **Guidance is hierarchical, providing full detail on all commands.**

Improve your GUI 1/2

- 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

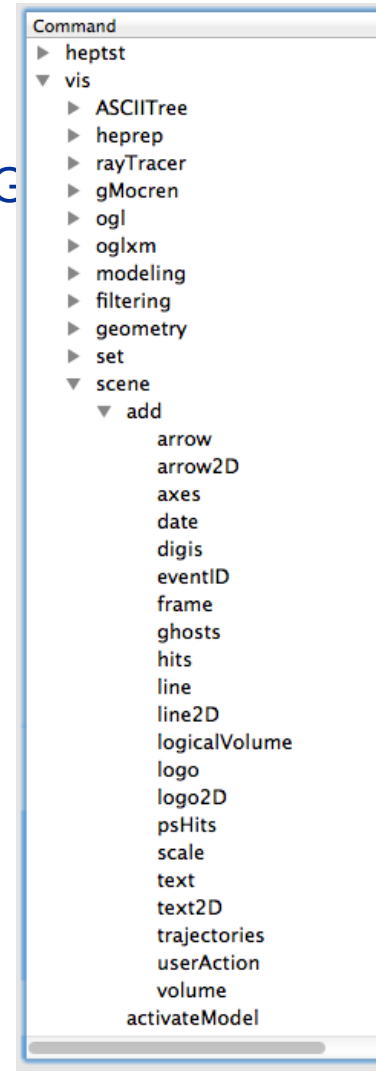
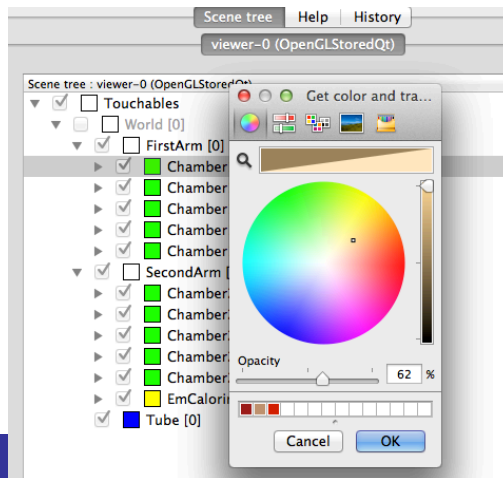
- User defined icons :

- `/gui/addIcon "Run beam on" user_icon "/run/beamOn 1" run.png`

Improve your GUI 2/2

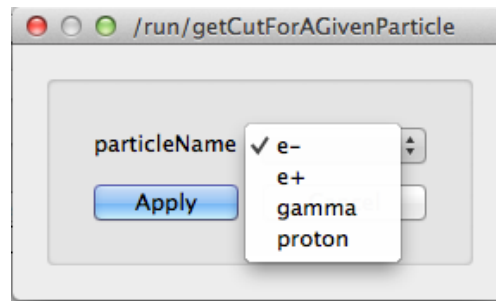
- Add extras :
 - Axes, Date, Scale, Text, Text2D...
- Export viewer in vectorial/non vectorial format (only for OpenGL)
 - Change size up to 8192*8192 (your max OpenGL card capacity)
 - `/vis/ogl/export <name.format> <width> <height>`
 - Available formats : eps/pdf/svg/ps, jpg, jp2, png, ...
- Change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

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

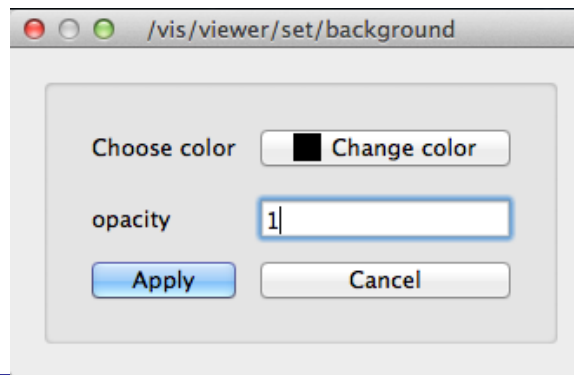


Improve your GUI, Qt special

- `/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"`



Time slicing

- **Already in Geant4 since 2006! (By J. Allison)**
- **Activate instantiation of G4RichTrajectory**
 - **ExN03TrackingAction::PreUserTrackingAction**
- **exampleB4 visTutor/exN03Vis12.mac**
 - a) Draw by charge with trajectory points
 - b) Draw by particle ID (remove γ 's)
 - c) π - μ -e decay
- **exampleB4 visTutor/exN03Vis13.mac**
 - 10 GeV EM shower showing light front
 - Camera follows (pans) at speed of light

Time slicing - Geant4 >=10.5

- **/vis/viewer/set/timeWindow/ directory**
- **Guidance**

For these commands use

```
/vis/scene/add/trajectories rich
```

```
/vis/modeling/trajectories/drawByCharge-0/default/setTimeSliceInterval 0.01 ns
```

then typically

```
/vis/viewer/set/timeWindow/displayLightFront true 0 0 -50 cm -0.5 ns
```

```
/vis/viewer/set/timeWindow/displayHeadTime true
```

```
/vis/viewer/set/timeWindow/fadeFactor 1
```

```
/run/beamOn # or several until you get a good event or events
```

```
/vis/viewer/set/timeWindow/startTime 0 ns 1 ns
```

```
/vis/viewer/save
```

```
/vis/viewer/set/timeWindow/startTime 1 ns 1 ns
```

then repeat with next start time, another view and a save, then try

```
/vis/viewer/interpolate
```

Time slicing - Geant4 <=10.4

- **/vis/ogl/set/ directory**
- **Guidance**

For these commands use

```
/vis/scene/add/trajectories rich  
/vis/modeling/trajectories/drawByCharge-0/default/setTimeSliceInterval 0.01 ns
```

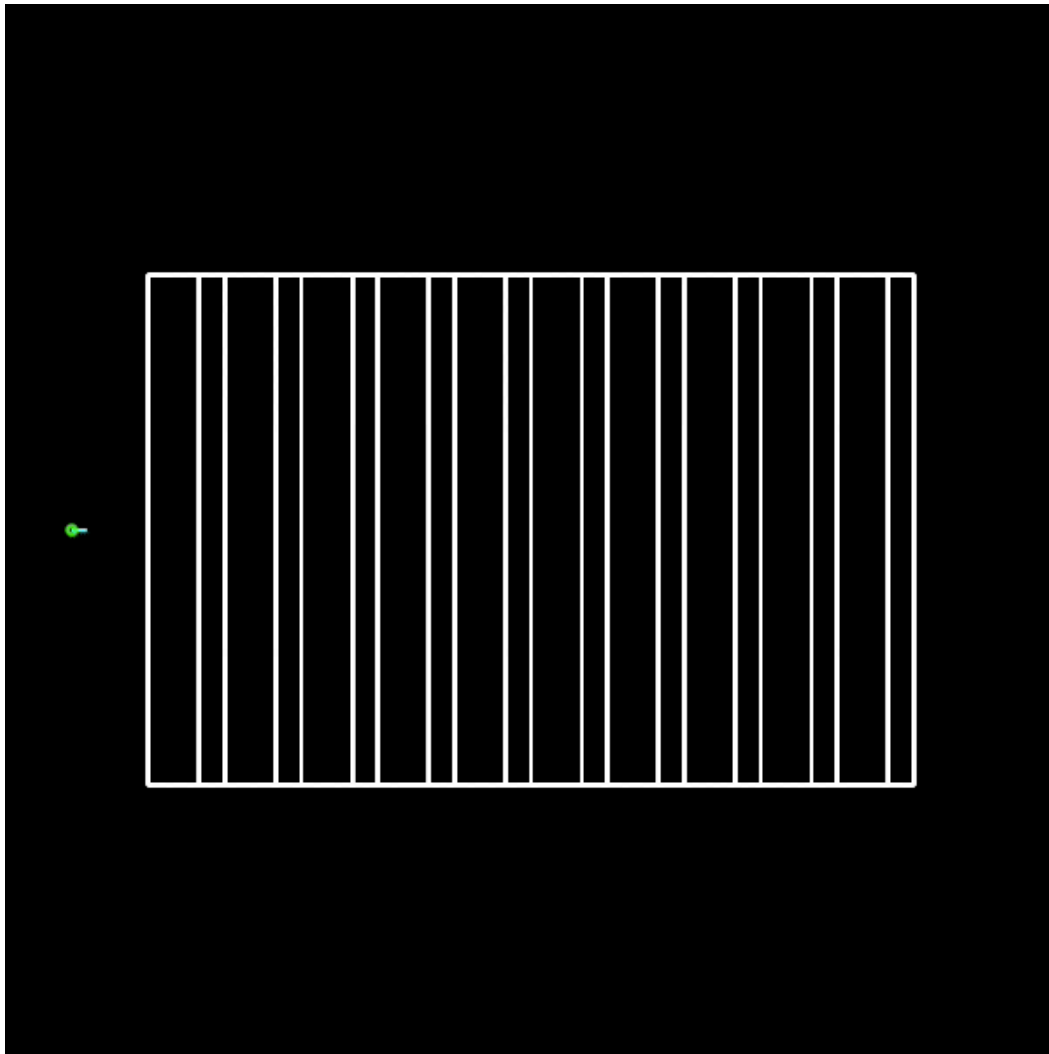
then typically

```
/vis/ogl/set/displayLightFront true 0 0 -50 cm -0.5 ns  
/vis/ogl/set/displayHeadTime true  
/vis/ogl/set/fade 1  
/run/beamOn # or several until you get a good event or events
```

```
/vis/ogl/set/startTime 0 ns 1 ns  
/vis/ogl/export  
/vis/ogl/set/startTime 1 ns 1 ns  
/vis/ogl/export  
...
```

then repeat with next start time, another view and a save...

Time slicing



N04 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

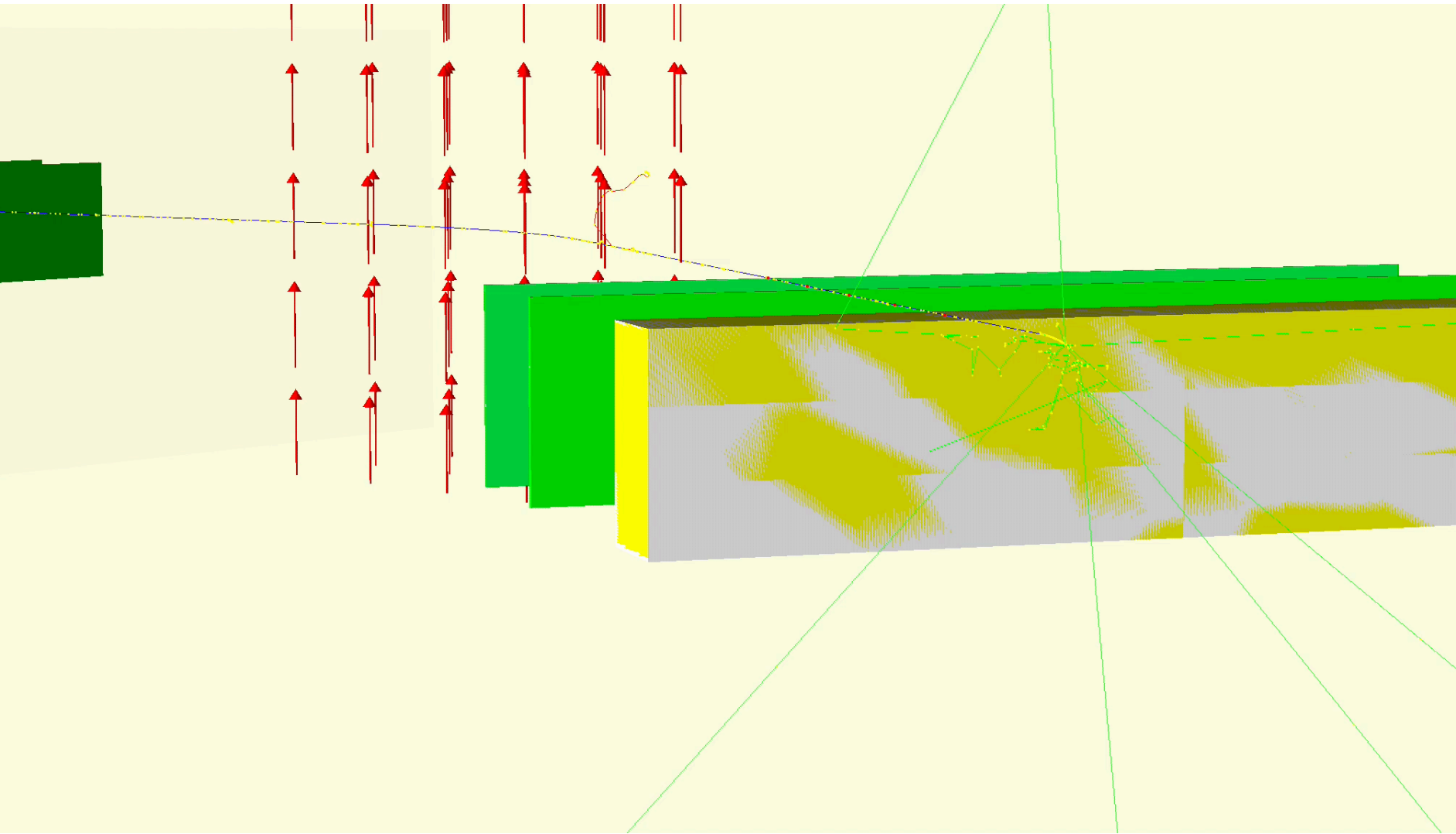
ν green

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



References

- **Geant4 Qt User Interface tutorial**
<http://geant4.in2p3.fr/spip.php?article84&lang=en>