GEANT4
A SIMULATION TOOLKIT

cnrs dépasser les frontières IN2P3 Les deux infinis

UNIVERSITÉ PARIS SUD
Comprendre le monde, construire l'avenir®

Scoring - 3

I. Hrivnacova, IJCLab Orsay

Credits M. Asai (SLAC), G. Folger (CERN) and others

Geant4 IN2P3 and ED PHENIICS Tutorial,
22 – 26 May 2023, IJCLab

Outline

- Geant4 scorers
- Command-based scoring
- Geant4 scorers & analysis

Geant4 Scorers

Ready to Use Scoring Classes

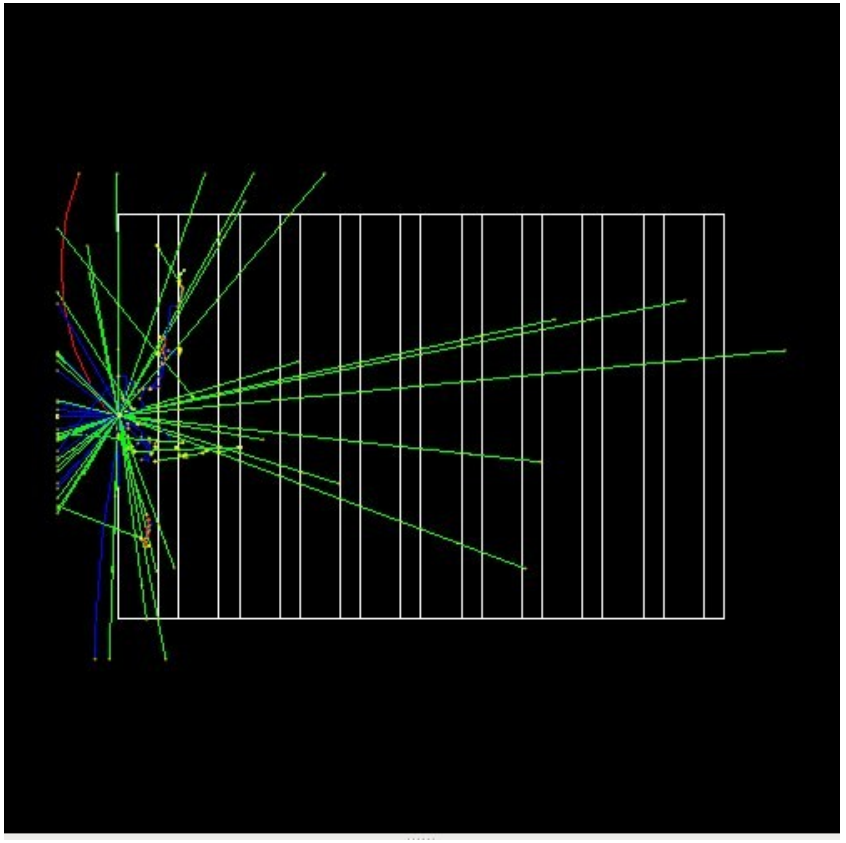
- Typical quantities, such as energy deposit, track length etc. can be accounted using the classes already available in Geant4
- There are three kinds of these classes
  - **"Primitive scorer"** – an object that accounts just one concrete quantity
    - G4VPrimitiveScorer base class
  - **"Filter"** – an object that applies a selection whether a quantity will be accounted: eg. the quantity will be counted only for charged particles
    - G4VSDFilter base class
    - Associated to a scorer
  - **G4MultiFunctionalDetector** - a sensitive detector that contains all scorers and delegate accounting (scoring) to them
    - G4VSensitiveDetector base class

Ready to Use Scoring Classes - 2

- Primitive scorer classes:
  - G4PSTrackLength, G4PSEnergyDepsit, G4PSDoseDeposit, G4PSChargeDeposit, G4PSFlatSurfaceCurrent, G4PSNofSecondary, G4PSNofStep, …
- Filter classes:
  - G4SDCharged[Neutral]Filter, G4SDParticleFilter, G4SDKineticEnergyFilter, …
- Users can also implement their own primitive scorer or filter class derived from Geant4 base class
- See Application Developers Guide - Detector Definition and Response - Hits for the complete list of scorers and filters

Example B4d

- An example of use of Geant4 scorers is provided in basic example B4d
- G4MultiFunctionalDetector
- Scorers accounting energy deposit and track length: G4PSEnergyDeposit, G4PSTrackLength
- Filter to select charged particles: G4SDCharchedFilter



Command-based scoring

Command-based scoring

- Command-based scoring functionality offers a built-in scoring mesh and various scorers for commonly-used physics quantities such as dose, flux, etc.
  - Due to small performance overhead, it does not come by default.
- To use this functionality, activate the G4ScoringManager after the instantiation of G4RunManager in your main()
- This will create the UI commands of this functionality in /score directory.

```
#include "G4ScoringManager.hh"
int main()
{
   // ...
   G4RunManager* runManager = new G4RunManager;
   G4ScoringManager::GetScoringManager();
   // ...
}
```

Command-based Scoring

Example Macro

```
# Define scoring mesh
/score/create/boxMesh boxMesh
/score/mesh/boxSize 100. 100. 100. cm
/score/mesh/nBin 30 30 30

# Define scoring quantity
/score/quantity/energyDeposit boxMeshPS keV

# Define a filter
/score/filter/charged boxMeshFilter

# Close mesh
/score/close
```

3D scoring mash:
name,
shape&size,
number of bins

Scoring quantity:
energyDeposit

Filter:
charged particles only

Close mash

Command-based Scoring

Drawing a Score

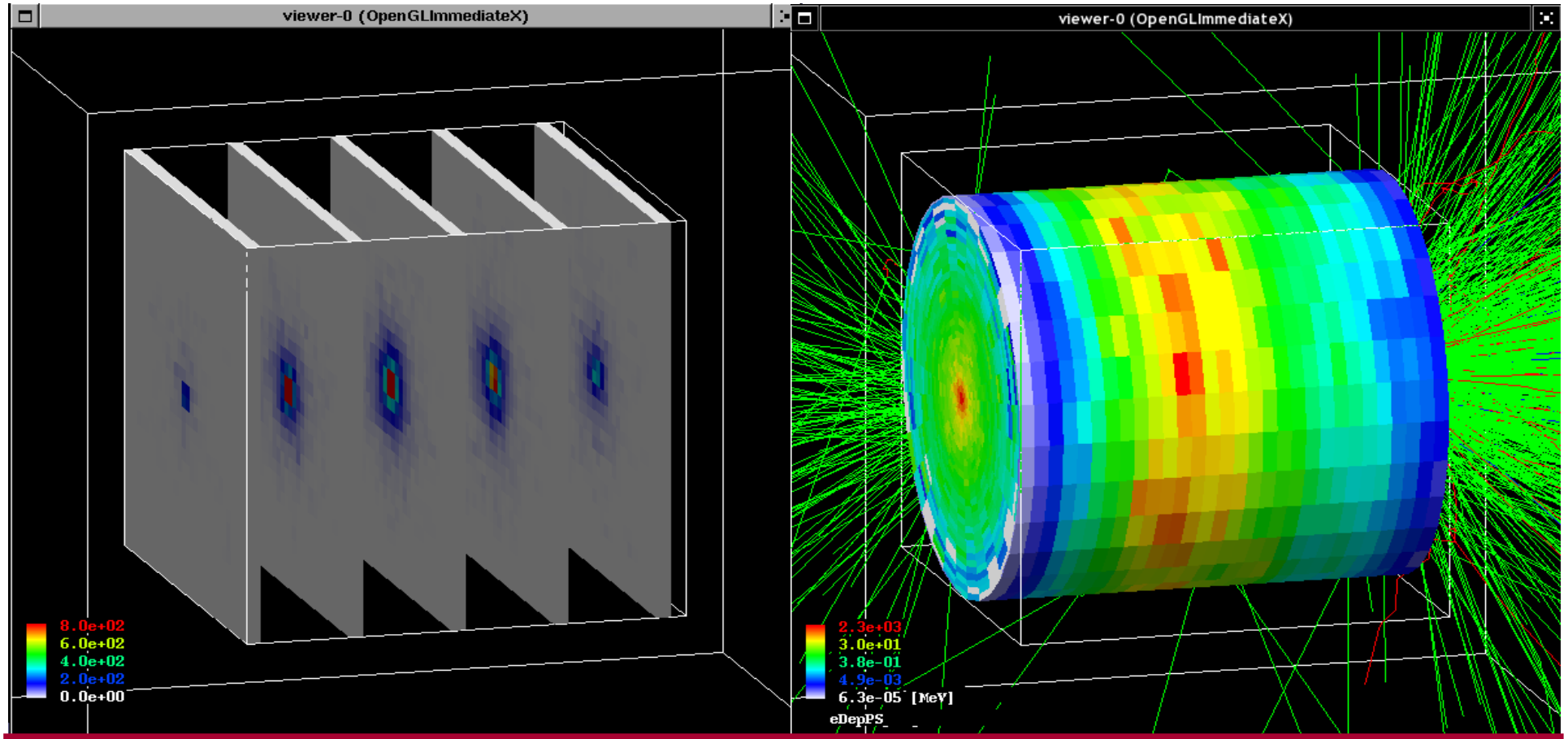
```
# Draw a projection
/score/drawProjection <mesh_name> <scorer_name> <color_map>

# Draw a slice
/score/drawColumn <mesh_name> <scorer_name> <plane> <column>
```

- Color map
  - By default, linear and log-scale color maps are available.
  - Minimum and maximum values can be defined by `/score/colorMap/setMinMax` command. Otherwise, min and max values are taken from the current score.

Command-based Scoring

Drawing a Score (2)



From M.Asai

Command-based Scoring

Write Scores To A File

```
# Single score
/score/dumpQuantityToFile <mesh_name> <scorer_name> <file_name>

# All scores
/score/dumpAllQuantitiesToFile <mesh_name> <file_name>
```

- By default, values are written in CSV.
- By creating a concrete class derived from G4VScoreWriter base class, the user can define his own file format.
  - Example in /examples/extended/runAndEvent/RE03
  - Users score writer class should be registered to G4ScoringManager.

Geant4 Scorers & Analysis

Score Ntuple Writer

- The scorers hits can be also saved using Geant4 analysis tools.
  - Demonstrated in **B3** and **B4d** basic examples
- Storing hits is activated in the main() function with instantiating G4TScoreNtupleWriter.

```
#include "G4AnalysisManager.hh"
#include "G4TScoreNtupleWriter.hh"
...
G4TScoreNtupleWriter<G4AnalysisManager> scoreNtupleWriter;
```

- The Geant4 UI commands can be used to choose the output file name and the level of verbosity

```
/score/fileName name
/score/writerVerbose 1
```

Summary

- The Geant4 toolkit provides dedicated classes/tools for user scoring:
  - Sensitive detectors
  - Geant4 scorers
  - Command-based scoring
  - Geant4 analysis tools