YAGA

yet another geometry approach

Benedikt Zihlmann

The Detector Geometry

- Geometry defined in
  `${HALLD_HOME\}/src/programs/Simulation/hdds`
- hdds-geant generates geometry hddsGeant3.F for the geant3 based Monte Carlo
- hdds-root generates geometry hddsroot.C for interactive root session.

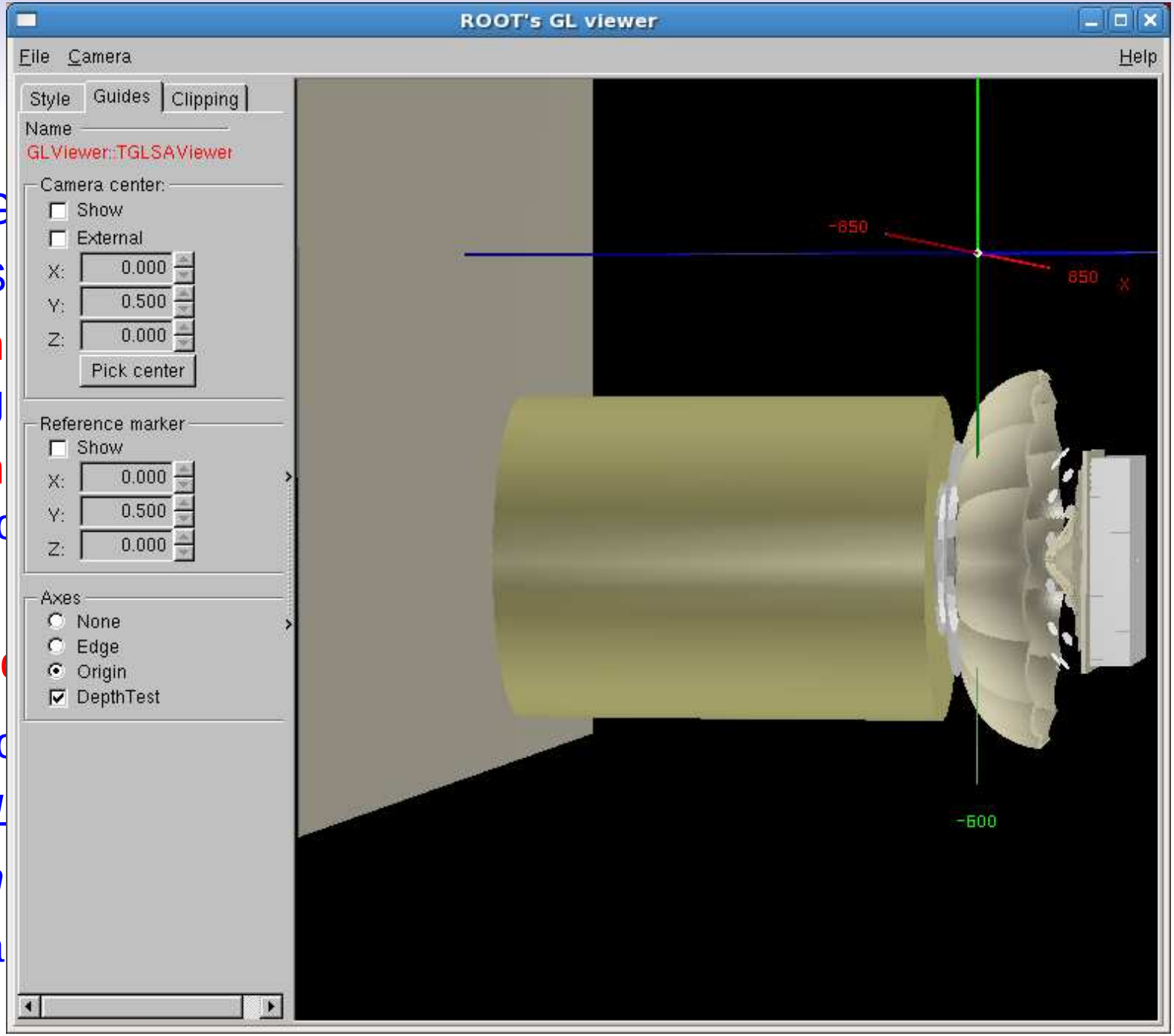
use hddsroot.C interactively in root:

- root
- *.L hddsroot.C*
- *hddsroot*()
- and use the gui:

- G
  $
- h
  g
- h
  rc

use hc

- rc
- *.L*
- *h*
- a



Use hddsroot.C in DANA

Modify hddsroot.C to hddsroot.cc in my private HDGEOMETRY

```
#include "DRootGeometry.h";
TGeoManager * hddsroot()
{
//
//  This file has been generated automatically via the
//  utility hdds-root directly from main_HDDS.xml
//   (see ROOT class TGeoManager for an example of use)
//
//gSystem->Load("libGeom");
//TGeoRotation *rot;
//TGeoNode *Node, *Node1;

//TGeoManager *detector = new TGeoManager("hddsroot","hddsroot.C");

//-----------List of Materials and Mixtures-------------
.
.
.
//Double_t *origin = new Double_t[3];
//origin[0] = 450; origin[1] = -50; origin[2] = -200;
//TGeoBBox *clip = new TGeoBBox("CLIP",300,300,300,origin);
//gGeoManager->SetClippingShape(clip);
//gGeoManager->DefaultColors();
//gGeoManager->SetVisLevel(9);
//HALL->Raytrace();
return gGeoManager;
}
```

The Header file

The header file DRootGeometry.h

```
#include <TROOT.h>
#include <TGeoManager.h>
#include <TGeoMaterial.h>
#include <TGeoMedium.h>
#include <TGeoVolume.h>
#include <TGeoPgon.h>
#include <TGeoMatrix.h>

TGeoManager* hddsroot();
```

USE hddsroot.cc and DRootGeometry.h in the DANA framework

Use hddsroot.cc in DANA

my small MyProcessor.cc DANA program

```
//
// MyProcessor.cc
//
#include <iostream>
using namespace std;
#include "HDGEOMETRY/DRootGeometry.h"
...
jerror_t MyProcessor::brun(JEventLoop *eventLoop, int runnumber) {
  TGeoManager *DGeom = hddsroot();
  Double_t TotRadLen = 0;
  DGeom->InitTrack(0.,0.,65.,0.,cos(85./180.*3.1415926),cos(5./180.*3.1415926));
  while (!DGeom->IsOutside()) {   // you should use this to propagate till outside
    TGeoNode *cnod = DGeom->FindNode();
    Double_t RadLen = cnod->GetMedium()->GetMaterial()->GetRadLen();
    DGeom->FindNextBoundary();
    Double_t len = DGeom->GetStep();
    const Double_t *cpoint = DGeom->GetCurrentPoint();

    if (RadLen==0) {
      cout<<"Error RadLen is zero! at point "<<*cpoint<<" "
          <<*(cpoint+1)<<" "<<*(cpoint+2)<<endl;
      cout<<"Geometry tree: "<<path<<endl;
    }
    if ((*(cpoint+2)<500)&&RadLen!=0.)
    TotRadLen += len/RadLen;
    DGeom->Step();
  }
  cout<<"Total Radiation Length: "<<TotRadLen<<endl;
  return NOERROR;
}
```

And it works (sort of)

the result:

```
Begin of run
Initialize geometry
Info in <TGeoManager::TGeoManager>: Geometry Geometry, default geometry created
Info in <TGeoManager::SetTopVolume>: Top volume is SITE. Master volume is SITE
Error in <TGeoMaterial::SetRadLen>: Invalid material (null): a=0.000000 z=0.000000 -> user va
Info in <TGeoManager::CheckGeometry>: Fixing runtime shapes...
Info in <TGeoManager::CheckGeometry>: ...Nothing to fix
Info in <TGeoManager::CloseGeometry>: Counting nodes...
Info in <TGeoManager::Voxelize>: Voxelizing...
Info in <TGeoManager::CloseGeometry>: Building cache...
Info in <TGeoNavigator::BuildCache>: --- Maximum geometry depth set to 100
Info in <TGeoManager::CloseGeometry>: 60774 nodes/ 374 volume UID's in default geometry
Info in <TGeoManager::CloseGeometry>: ----------------modeler ready----------------
X= 0    Y=0    Z=65.
Name:     LIH2
Density:  0.0708
RadLen:   816.933
Total Radiation Length: 0.132619
```

It works but: Not all Materials have a radiation length set.
`Material_HDDS.xml` needs to be checked.
hdds-root.cpp already modified to get radiation length for Vacuum correct;