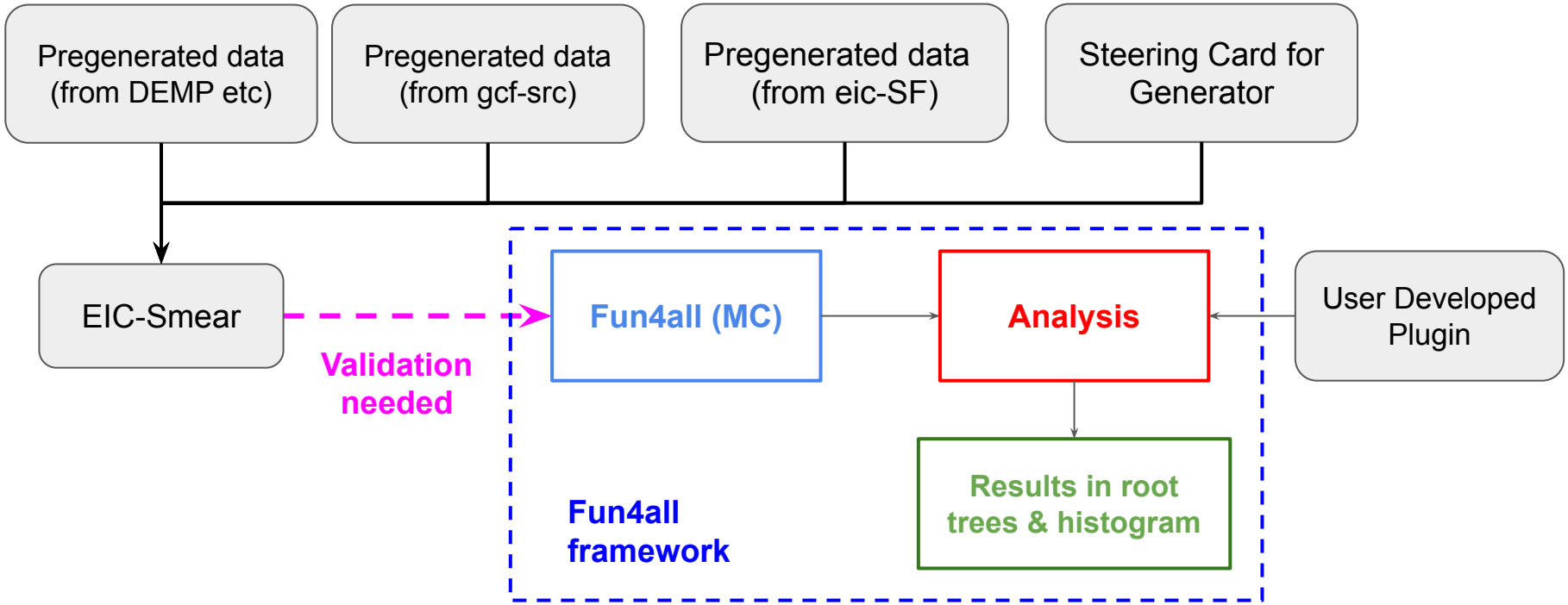
# ECCE Simulation Tutorial: Analysis Example on Diffractive and Tagging Working Group

**Wenliang (Bill) Li (W&M)**

**20/May/2020**

WILLIAM & MARY
CHARTERED 1693

# Fun4all (full geant4) Workflow for Our Group



# A Brute Force Example Tutorial (No why, only how)

- Creating a diffractive and tagging group oriented analysis module
- How to master particle gun from the
  - Single particle
  - Sample of particles
- Validating your generator output
- Modifying the ZDC

# Objective

- **Create a step-by-step guide**

Execute the **Red command line** sequentially

# Prerequisites

- **Virtualbox:**
  - **Instruction:** https://github.com/ECCE-EIC/Singularity/blob/master/VirtualBox.md
- **EIC Tutorial macro:**
  - **git clone https://github.com/ECCE-EIC/macros**
- **Analysis plugin tutorial scripts:**
  - **git clone https://github.com/ECCE-EIC/tutorials**
- **Example script and data samples from Bill:**
  - **git clone https://github.com/billlee77/bill_diff_tagg_script**

# To Start

- Source correct environment and setting up our workspace:
  - Fresh terminal in VM
  - cd ~/
  - source singularity_shell.sh
  - source setup.sh
  - mkdir work
  - cd work
  - git clone https://github.com/ECCE-EIC/macros
  - git clone https://github.com/ECCE-EIC/tutorials
  - git clone https://github.com/billlee77/bill_diff_tagg_script

# Creating Diffractive and Tagging Analysis Module!

- Still under ~/work/ directory:
  - mkdir diff_tagg_ana
  - cd diff_tagg_ana
  - CreateSubsysRecoModule.pl diff_tagg_ana

```
Singularity> CreateSubsysRecoModule.pl diff_tagg_ana
Singularity> ls
diff_tagg_ana.cc  diff_tagg_ana.h
Singularity>
```

  - cp ~/work/tutorials/CaloAna/src/*.[sa]* .
  - ls

```
autogen.sh  configure.ac  diff_tagg_ana.cc  diff_tagg_ana.h  Makefile.am
```

# Modify Makefile.am & Compile

- Modify as shown

```
AUTOMAKE_OPTIONS = foreign

lib_LTLIBRARIES = \
  libcaloana.la

AM_LDFLAGS = \
  -L$(libdir) \
  -L$(OFFLINE_MAIN)/lib

AM_CPPFLAGS = \
  -I$(includedir) \
  -I$(OFFLINE_MAIN)/include \
  -I$(ROOTSYS)/include

pkginclude_HEADERS = \
  CaloAna.h

if ! MAKEROOT6
  ROOT5_DICTS = \
    CaloAna_Dict.cc
endif

libcaloana_la_SOURCES = \
  $(ROOT5_DICTS) \
  CaloAna.cc

libcaloana_la_LDFLAGS = \
  -L$(libdir) \
  -L$(OFFLINE_MAIN)/lib \
  -lcalo_io \
  -lfun4all \
  -lg4detectors_io \
  -lphg4hit


##############################################
# linking tests

noinst_PROGRAMS = \
  testexternals

testexternals_SOURCES = testexternals.C
testexternals_LDADD = libcaloana.la
```

libdifftaggana.la

diff_tagg_ana.h

diff_tagg_ana_Dict.h

libdifftaggana_la_SOURCES

diff_tagg_ana.cc

libdifftaggana_la_LDFLAGS

libdifftaggana.la

we will cheat

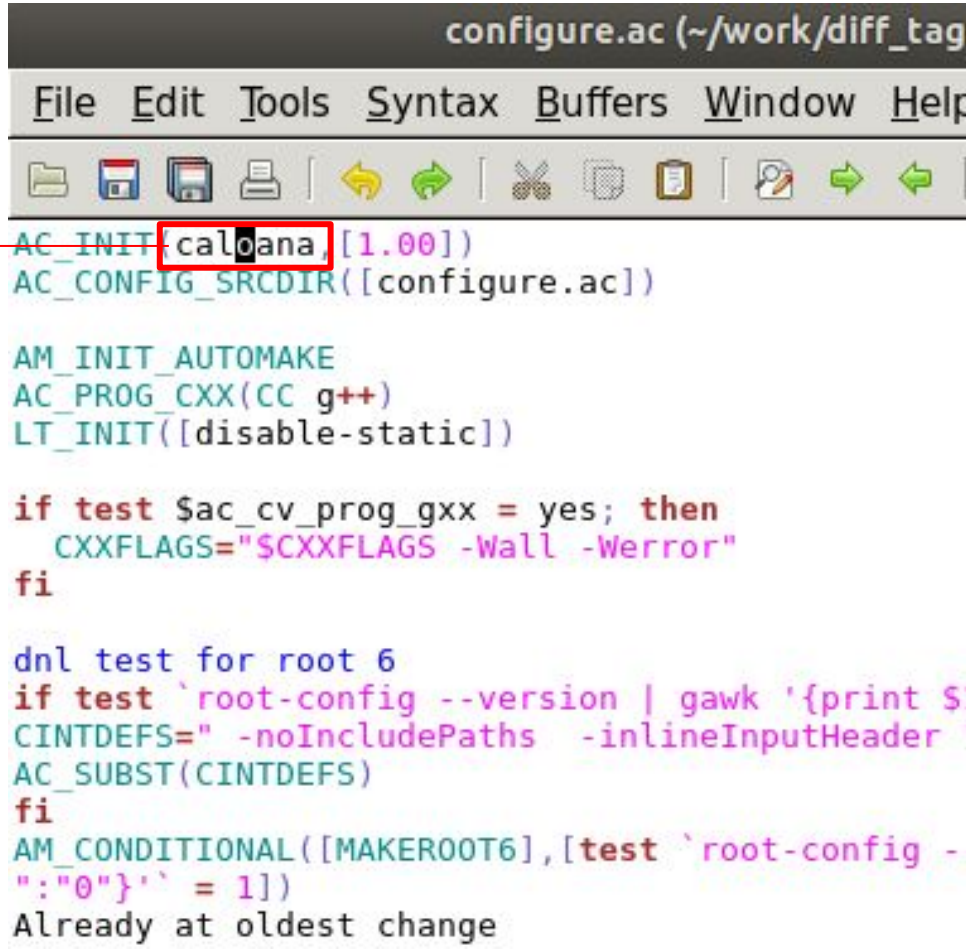
- cp ~/work/bill_diff_tagg_script/diff_tagg_ana/Makefile.am .

# Modify configure.ac & Compile

we will cheat again

cp ~/work/bill_diff_tagg_script/diff_tagg_ana/configure.ac .

difftaggana



# Compile and check

- 
- chmod +x autogen.sh
- mkdir build
- cd build
- ../autogen.sh --prefix=$MYINSTALL
- make install

- 
```
Singularity> ls ~/install/lib/
libdifftaggana.la     libdifftaggana.so.0.0.0  libdifftaggmodule.so.0
libdifftaggana.so     libdifftaggmodule.la     libdifftaggmodule.so.0.0.0
libdifftaggana.so.0   libdifftaggmodule.so
```

- ls ~/install/include/difftaggana/

```
Singularity> ls ~/install/include/difftaggana/
diff_tagg_ana.h
```

# What is in the plugin?

## Inside diff_tagg_ana.cc

### Initialization

```
//____________________________________________________________..
int diff_tagg_ana::Init(PHCompositeNode *topNode)
{

  hm = new Fun4AllHistoManager(Name());
  // create and register your histos (all types) here
  // TH1 *h1 = new TH1F("h1",....)
  // hm->registerHisto(h1);
  outfile = new TFile(outfilename.c_str(), "RECREATE");
  g4hitntuple = new TNtuple("hitntup", "G4Hits", "x0:y0:z0:x1:y1:z1:edep");

  std::cout << "diff_tagg_ana::Init(PHCompositeNode *topNode) Initializing" << std::endl;
  return Fun4AllReturnCodes::EVENT_OK;
}
```

### looping over events

```
//____________________________________________________________
int diff_tagg_ana::process_event(PHCompositeNode *topNode)
{
  std::cout << "diff_tagg_ana::process_event(PHCompositeNod

  process_g4hits(topNode);

  return Fun4AllReturnCodes::EVENT_OK;
}
```

### looping over all ZDC hits

```
int diff_tagg_ana::process_g4hits(PHCompositeNode* topNode)
{
  ostringstream nodename;

  // loop over the G4Hits
  nodename.str("");
//  nodename << "G4HIT_" << detector;
//  nodename << "G4HIT_" << "ZDC";
  nodename << "G4HIT_" << "ZDC";
//  nodename << "G4HIT_" << "EEMC";

  PHG4HitContainer* hits = findNode::getClass<PHG4HitContainer>(topNode, nodename.str().c_str());

  cout << "Which detector ???  ???   " << nodename.str().c_str() << endl;

  if (hits)
  {
    // this returns an iterator to the beginning and the end of our G4Hits
    PHG4HitContainer::ConstRange hit_range = hits->getHits();
    for (PHG4HitContainer::ConstIterator hit_iter = hit_range.first; hit_iter != hit_range.second; hit_iter++)

    {

      cout << "AAA" << endl;

      // the pointer to the G4Hit is hit_iter->second
      g4hitntuple->Fill(hit_iter->second->get_x(0),
                        hit_iter->second->get_y(0),
                        hit_iter->second->get_z(0),
                        hit_iter->second->get_x(1),
                        hit_iter->second->get_y(1),
                        hit_iter->second->get_z(1),
                        hit_iter->second->get_edep());

//      hit_iter->get_avg_t();

    }
  }

  cout << "BB" << endl;

  return Fun4AllReturnCodes::EVENT_OK;
}
```

# Modify plugin with ZDC Information and recompile

Load the prepared analysis plugin

- cp ~/work/bill_diff_tagg_script/diff_tagg_ana/diff_tagg_ana.cc ../
- cp ~/work/bill_diff_tagg_script/diff_tagg_ana/diff_tagg_ana.h ../
- make install

Make install is needed after every change made to analysis plugin.

# Let use the analysis module Enable G4User Function

Let go to the main script:

- cd ~/work/macros/detectors/EICDetector/
- cp ~/work/bill_diff_tagg_script/main_macro/G4_User.C .

**All setup is completed !**

In main macron: Fun4All_G4_EICDetector.C

Turning on USER defined function

```
Enable::USER = true;
```

Line 346

# Particle Gun!

- **Single gun line 160**

```
if (Input::GUN)
{
  INPUTGENERATOR::Gun[0]->AddParticle("pi0", 50*sin(0.025), 0, 50*cos(0.025));
  INPUTGENERATOR::Gun[0]->set_vtx(0, 0, 0);
}
```

Test with a 50 GeV $\pi^0$ toward ZDC (25 mRad)

- **Turn on the gun option**

```
Input::GUN = true;
```
Line 89

- **Turn off SIMPLE Generator**

```
//Input::SIMPLE = true;
```
Line 84

**Where Pb can be created**

**Doxygen**

**IONGUN Class**

**PHG4IonGun Class Reference**

`#include <fun4all_coresoftware/blob/master/simulation/g4sim`

▸ Inheritance diagram for PHG4IonGun:

▸ Collaboration diagram for PHG4IonGun:

Public Member Functions

| | |
|---|---|
| | PHG4IonGun (const std::string &name="PHG4IONGUN") |
| virtual | ~PHG4IonGun () |
| int | process_event (PHCompositeNode *topNode) |
| void | SetA (const int a) |
| void | SetZ (const int z) |
| void | SetCharge (const int c) |
| void | ExcitEnergy (const double e) |
| void | SetMom (const double px, const double py, const double pz) |
| void | Print (const std::string &what="ALL") const |

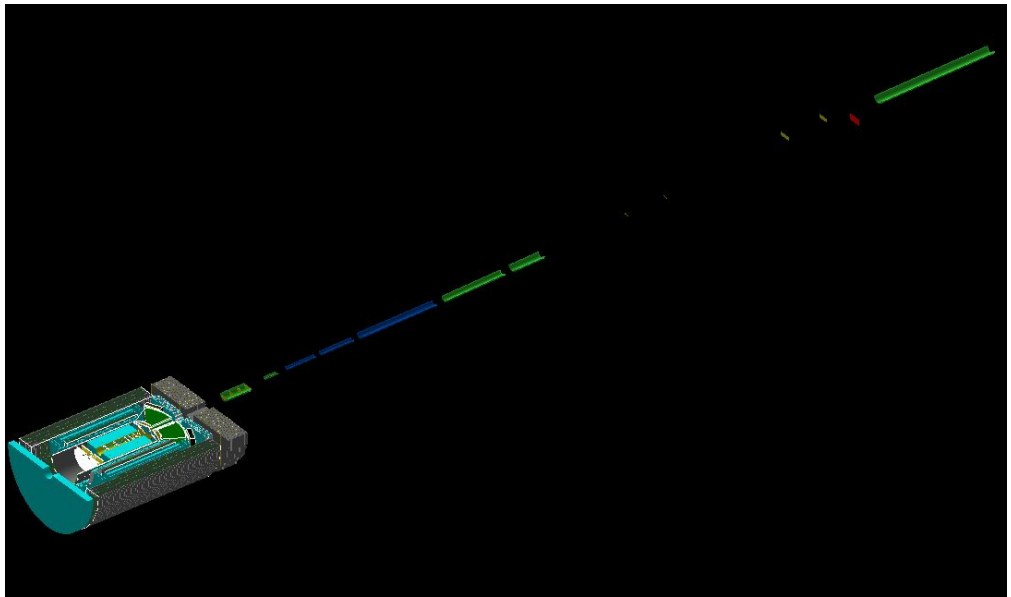
- **cp ~/work/bill_diff_tagg_script/main_macro/Fun4All_G4_EICDetector_GUN.C  Fun4All_G4_EICDetector.C**

# Particle Gun Visual Validation

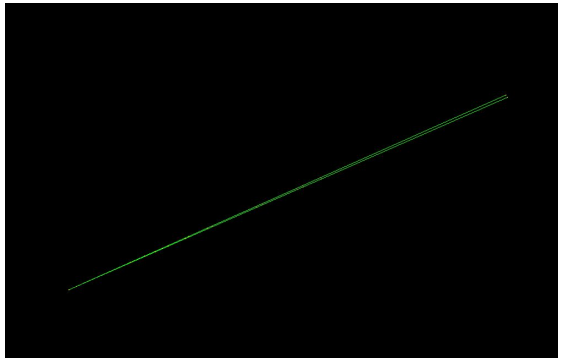
- Turning on Display

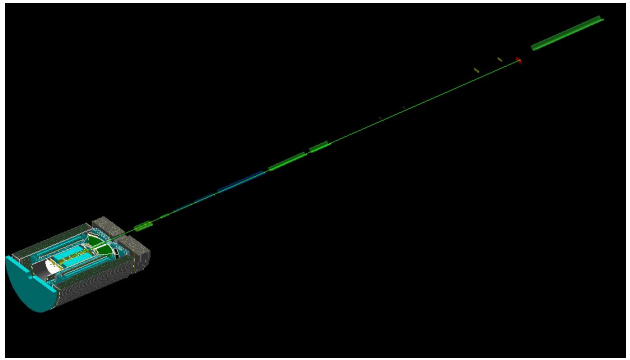
```
Enable::DISPLAY = true;
```
Line 232

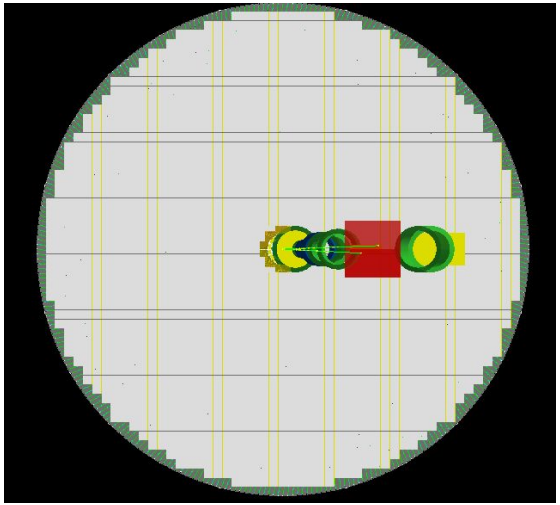
- root Fun4All_G4_EICDetector.C



# Particle Gun Visual Validation







se->run(1)

g4->ApplyCommand("/vis/viewer/refresh")

**Where did detector go?**

**Refresh needed here**

g4->ApplyCommand("/vis/viewer/set/viewpointThetaPhi 0 0")

g4->ApplyCommand("/vis/viewer/zoom 4")

g4->ApplyCommand("/vis/viewer/addCutawayPlane 0 3 0 m 0 -1 0")

# One should also see some wired outputs : )

```
diff_tagg_ana::process_event(PHCompositeNode *topNode) Processing Event
Which detector ???  ???   G4HIT_ZDC
BB
diff_tagg_ana::ResetEvent(PHCompositeNode *topNode) Resetting internal structure
s, prepare for next event
diff_tagg_ana::EndRun(const int runnumber) Ending Run for Run 0
diff_tagg_ana::End(PHCompositeNode *topNode) This is the End...
Fun4AllHistoManager::dumpHistos() Writing root file: out.root
All done
diff_tagg_ana::Reset(PHCompositeNode *topNode) being Reset
diff_tagg_ana::~diff_tagg_ana() Calling dtor
```

# Multiple particles in certain angular-momentum range

- Using Simple Generator (default) within Fun4All_G4_EICDetector.C
  - turning off GUN and turn on simple

```
// Simple Input generator:
// if you run more than one of these Input::SIMPLE_NUMBER > 1
// add the settings for other with [1], next with [2]...
if (Input::SIMPLE)
{
  INPUTGENERATOR::SimpleEventGenerator[0]->add_particles("pi-", 5);
  if (Input::HEPMC || Input::EMBED)
  {
    INPUTGENERATOR::SimpleEventGenerator[0]->set_reuse_existing_vertex(true);
    INPUTGENERATOR::SimpleEventGenerator[0]->set_existing_vertex_offset_vector(0.0, 0.0, 0.0);
  }
  else
  {
    INPUTGENERATOR::SimpleEventGenerator[0]->set_vertex_distribution_function(PHG4SimpleEventGenerator::Uniform,
                                                                              PHG4SimpleEventGenerator::Uniform,
                                                                              PHG4SimpleEventGenerator::Uniform);
    INPUTGENERATOR::SimpleEventGenerator[0]->set_vertex_distribution_mean(0., 0., 0.);
    INPUTGENERATOR::SimpleEventGenerator[0]->set_vertex_distribution_width(0., 0., 5.);
  }
  INPUTGENERATOR::SimpleEventGenerator[0]->set_eta_range(-3, 3);
  INPUTGENERATOR::SimpleEventGenerator[0]->set_phi_range(-M_PI, M_PI);
  INPUTGENERATOR::SimpleEventGenerator[0]->set_pt_range(0.1, 20.);
}
```

- cp ~/work/bill_diff_tagg_script/main_macro/Fun4All_G4_EICDetector_GUN.C  Fun4All_G4_EICDetector.C

# Validating data from the generator

**Sample data stored in ~/work/bill_diff_tagg_script/sample_data**

**Turning data to fun4all compatible tree:**

- cd ~/work/bill_diff_tagg_script/sample_data/
- root -l
- gSystem->Load("libeicsmear.so")
- BuildTree("eic_input_DEMPGen_Pion0_no_decay_validation.dat", ".", -1, "log.txt")
- .q

**Validate the generated data:**

- cd ~/work/bill_diff_tagg_script/sample_data/
- cd ~/work/macros/detectors/EICDetector
- cp ~/work/bill_diff_tagg_script/main_macro/Fun4All_G4_EICDetector_validation.C  Fun4All_G4_EICDetector.C

# Loading you own data

In Fun4All_G4_EICDetector.C:

```
    const string &inputFile = "/home/fun4all/work/bill_diff_tagg_script/sample_d
ata/eic_input_DEMPGen_Pion0_no_decay_validation.root",
```

Line 28

Turn on READEIC

```
// eic-smear output
Input::READEIC = true;
INPUTREADEIC::filename = inputFile;
```

Line 101

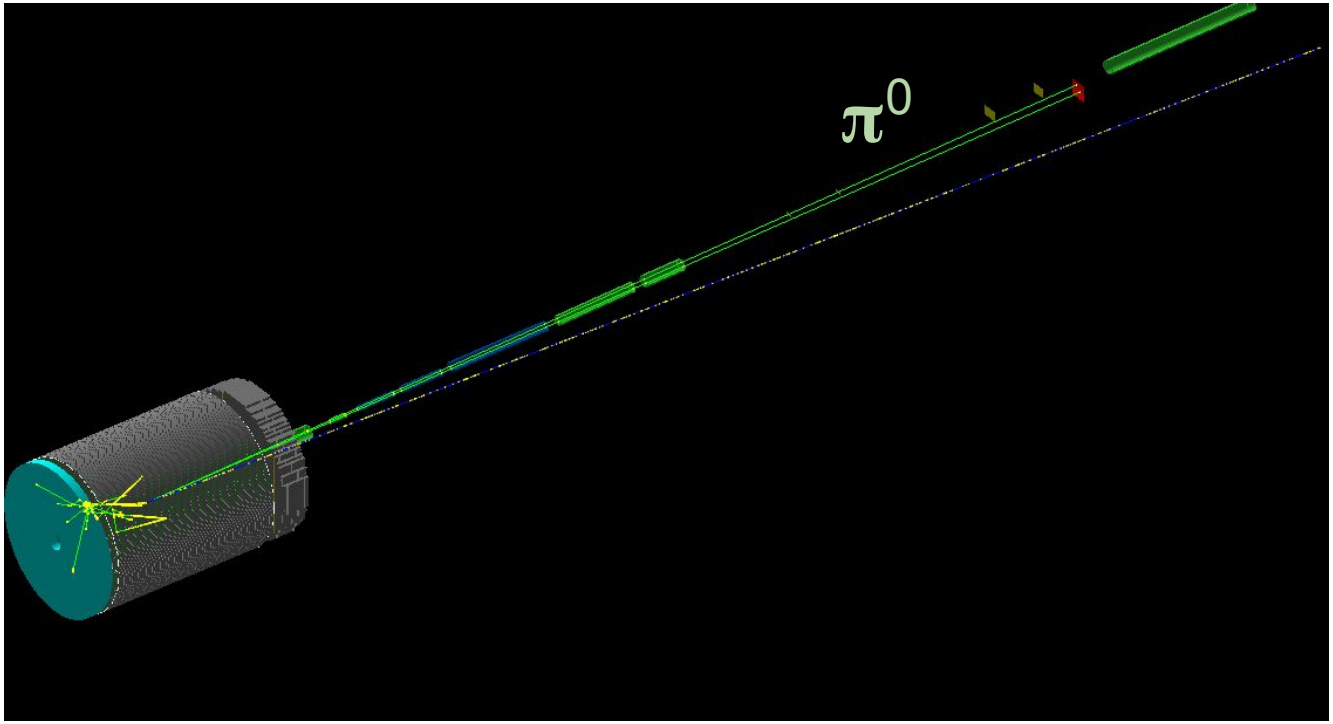
Turn off GUN and SIMPLE

```
//Input::SIMPLE = true;
// Input::GUN = true;
```

Line 84

Line 89

# Validation




- root Fun4All_G4_EICDetector.C
- se->run(1)
- g4->ApplyCommand("/vis/viewer/refresh")

# Why am I so excited to see the events?

- In DEMP generator, we simulated $e+p \rightarrow e+p+\pi^0$ in head-on collision
  - crossing angle of 0

- Simulation results confirm that data format and boost rotation algorithm in Fun4All_G4_EICDetector.C works!

```
// Reads event generators in EIC smear files, which is registered in InputRegister
if (Input::READEIC)
{
  //! apply EIC beam parameter following EIC CDR
  Input::ApplyEICBeamParameter(INPUTGENERATOR::EICFileReader);
}
```

Line 213

# How to modify a IP6 ZDC?

- ZDC defined under: ~/work/macros/common/G4_hFarFwdBeamLine_EIC.C

```
void hFarFwdDefineDetectorsIP6(PHG4Reco* g4Reco){

  if (Enable::HFARFWD_VIRTUAL_DETECTORS_IP6 && Enable::HFARFWD_VIRTUAL_DETECTORS_IP8)
  {
    cout << "You cannot have detectors enabled for both IP6 and IP8 ON at the same time" << endl;
    gSystem->Exit(1);
  }

  int verbosity = std::max(Enable::VERBOSITY, Enable::HFARFWD_VERBOSITY);

  auto *detZDC = new PHG4BlockSubsystem("zdcTruth");
  detZDC->SuperDetector("ZDC");
  detZDC->set_double_param("place_x",96.24);
  detZDC->set_double_param("place_y",0);
  detZDC->set_double_param("place_z",3750);
  detZDC->set_double_param("rot_y",-0.025*TMath::RadToDeg());
  detZDC->set_double_param("size_x",60);
  detZDC->set_double_param("size_y",60);
  detZDC->set_double_param("size_z",0.1);
  detZDC->set_string_param("material","G4_Si");
  detZDC->SetActive();
  detZDC->set_color(1,0,0,0.5);
  detZDC->BlackHole();
  if(verbosity)
    detZDC->Verbosity(verbosity);
  g4Reco->registerSubsystem(detZDC);
```

ZDC should be Red

Black hole

**We want to change it to blue**

# How to modify a IP6 ZDC?

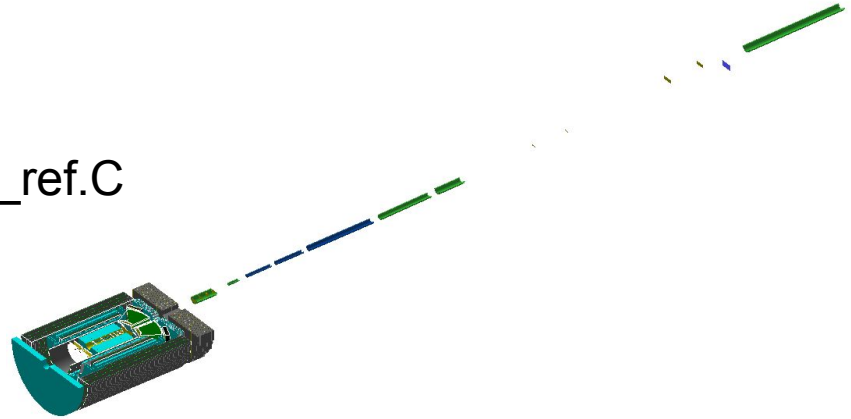
- cd ~/work/macros/detectors/EICDetector
- cp ~/work/macros/common/G4_hFarFwdBeamLine_EIC.C .

```
detZDC->set_color(1,0,0,0.5);  ⟶  detZDC->set_color(0,0,1,0.5);
```

- cp ~/work/bill_diff_tagg_script/main_macro/G4_hFarFwdBeamLine_EIC.C .
- root Fun4All_G4_EICDetector.C
- g4->ApplyCommand("/vis/viewer/set/background white")

Make ZDC as complicated as HCal?

see HCAL ~/work/macros/common/G4_HcalOut_ref.C



# Thanks you !

**If I can do it, you could too!**

**Any questions?**

# Generator output and validation

- **Standard format for EIC-smear does not include legendary LUND format:**
  - **Example below $e + p \rightarrow e' + p' + \pi^0$**

Standard for any e+p process don't change the order

```
SIMPLE Event FILE
=========================================
I, ievent, nParticles
=========================================
I  K(I,1)  K(I,2)  K(I,3)  K(I,4)  K(I,5)  P(I,1)  P(I,2)  P(I,3)  P(I,4)  P(I,5)  V(I,1)  V(I,2)  V(I,3)
=========================================
0          0          1
=========================================
1    21    11     0     3     4  6.12323e-16       0      -5       5    0.000511     0     0     0
2    21  2212     0     5     6           0       0     100     100.004   0.93827     0     0     0
3    21    22     1     0     0    -2.84037       0  -0.454462   -0.36    -2.85388     0     0     0
4     1    11     1     0     0     2.84037       0   -4.54554    5.36    0.000511     0     0     0
5     1  2212     2     0     0    -2.84037       0    45.282    45.3807   0.93827     0     0     0
6     1   111     2     0     0           0       0    54.2636   54.2637   0.134977     0     0     0
=============== Event finished ===============
```

  - **Reference :**
    - https://github.com/eic/eicsmeardetectors/blob/master/tests/simple_gen.txt
    - **Pythia6 format:** https://eic.github.io/software/pythia6.html#output-file-structure

- **Beagle format is supported.**

# Virtualbox is slow ?  (Nvidia Driver Issue?)

- /cvmfs on OpenScienceGrid update every one
  - Time to load new changes
- **Experiment with Display setting**
- **Turning DISPLAY off after validation process completes**
- **Running singularity on local machine or farm**
  - https://github.com/ECCE-EIC/Singularity#option-1-mount-eic-cvmfs
  - Option-2 Download the EIC Fun4All build via HTTPS archive
  - JLab ifarm compatible

```
Singularity> root -l Fun4All_G4_EICDetector.C
root [0]
Processing Fun4All_G4_EICDetector.C...
```

**Takes a long while to load**