Geant4

Lecture 5: Other stuff

The Source
Scoring mesh

The source

Defined in PrimaryGeneratorAction

One of the 3 compulsory classes. (The others are DetectorConstruction and PhysicsList. It is an Action invoked by **`RunManager->SetUserAction`**)

Needs a :

- constructor - create particle gun or source and set parameters
- destructor - delete gun/source
- action - fire gun

Generator constructor/ destructor in exampleB1

```
class B1PrimaryGeneratorAction : public G4VUserPrimaryGeneratorAction
{
  public:
    B1PrimaryGeneratorAction();
    virtual ~B1PrimaryGeneratorAction();

    // method from the base class
    virtual void GeneratePrimaries(G4Event*);

    // method to access particle gun
    const G4ParticleGun* GetParticleGun() const { return fParticleGun; }

  private:
    G4ParticleGun*  fParticleGun; // pointer a to G4 gun class
    G4Box* fEnvelopeBox;
};
```

.hh

ctor

dtor

Data members

override base-class function

specifc function

```
B1PrimaryGeneratorAction::B1PrimaryGeneratorAction()
: G4VUserPrimaryGeneratorAction(),
  fParticleGun(0),
  fEnvelopeBox(0)
{
  G4int n_particle = 1;
  fParticleGun  = new G4ParticleGun(n_particle);

  // default particle kinematic
  G4ParticleTable* particleTable = G4ParticleTable::GetParticleTable();
  G4String particleName;
  G4ParticleDefinition* particle
    = particleTable->FindParticle(particleName="gamma");
  fParticleGun->SetParticleDefinition(particle);
  fParticleGun->SetParticleMomentumDirection(G4ThreeVector(0.,0.,1.));
  fParticleGun->SetParticleEnergy(6.*MeV);
}

//....oooOOOOOooo.....oooOOOOOooo.....oooOOOOOooo.....oooOOOOO

B1PrimaryGeneratorAction::~B1PrimaryGeneratorAction()
{
  delete fParticleGun;
}
```

.cc

Doing the stuff

Gun has had particle type, energy and direction set in ctor. May have been modified by user interface

This is all about choosing the starting point so

```
void B1PrimaryGeneratorAction::GeneratePrimaries(G4Event* anEvent)
{
  //this function is called at the begining of each event
  //

  // In order to avoid dependence of PrimaryGeneratorAction
  // on DetectorConstruction class we get Envelope volume
  // from G4LogicalVolumeStore.

  G4double envSizeXY = 0;
  G4double envSizeZ = 0;

  if (!fEnvelopeBox)
  {
    G4LogicalVolume* envLV
      = G4LogicalVolumeStore::GetInstance()->GetVolume("Envelope");
    if ( envLV ) fEnvelopeBox = dynamic_cast<G4Box*>(envLV->GetSolid());
  }

  if ( fEnvelopeBox ) {
    envSizeXY = fEnvelopeBox->GetXHalfLength()*2.;
    envSizeZ = fEnvelopeBox->GetZHalfLength()*2.;
  }
  else  {
    G4ExceptionDescription msg;
    msg << "Envelope volume of box shape not found.\n";
    msg << "Perhaps you have changed geometry.\n";
    msg << "The gun will be place at the center.";
    G4Exception("B1PrimaryGeneratorAction::GeneratePrimaries()",
     "MyCode0002",JustWarning,msg);
  }

  G4double size = 0.8;
  G4double x0 = size * envSizeXY * (G4UniformRand()-0.5);
  G4double y0 = size * envSizeXY * (G4UniformRand()-0.5);
  G4double z0 = -0.5 * envSizeZ;

  fParticleGun->SetParticlePosition(G4ThreeVector(x0,y0,z0));

  fParticleGun->GeneratePrimaryVertex(anEvent);
}
```

The gun and the User Interface

At the `Idle>` prompt type `help`

Then type 14

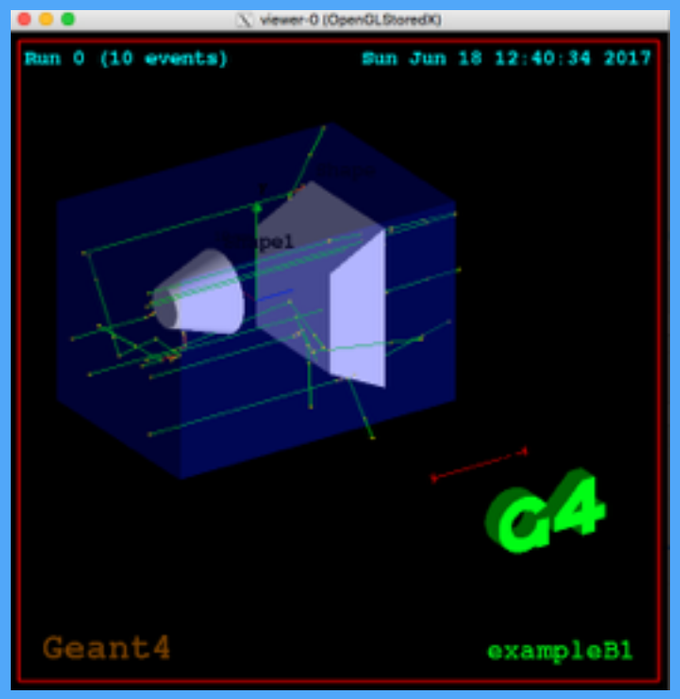
explore to see what's possible

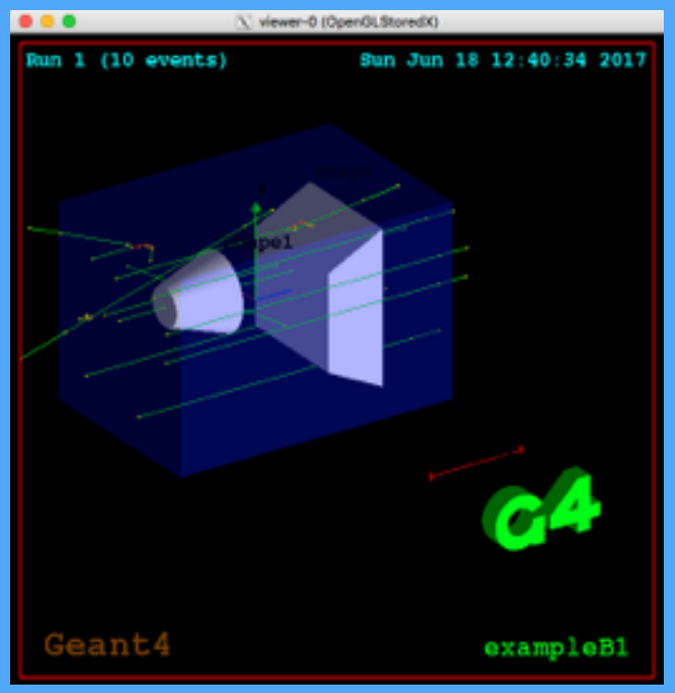
```
Idle> help
Command directory path : /
 Sub-directories :
 1) /control/   UI control commands.
 2) /units/   Available units.
 3) /process/   Process Table control commands.
 4) /analysis/   ...Title not available...
 5) /particle/   Particle control commands.
 6) /geometry/   Geometry control commands.
 7) /tracking/   TrackingManager and SteppingManager control commands.
 8) /event/   EventManager control commands.
 9) /cuts/   Commands for G4VUserPhysicsList.
 10) /run/   Run control commands.
 11) /random/   Random number status control commands.
 12) /material/   Commands for materials
 13) /physics_lists/   commands related to the physics simulation engine.
 14) /gun/   Particle Gun control commands.
 15) /vis/   Visualization commands.
 16) /heptst/   Controls for the hadronic energy/momentum test
 17) /physics_engine/   ...Title not available...
 Commands :

Type the number ( 0:end, -n:n level back ) :
```

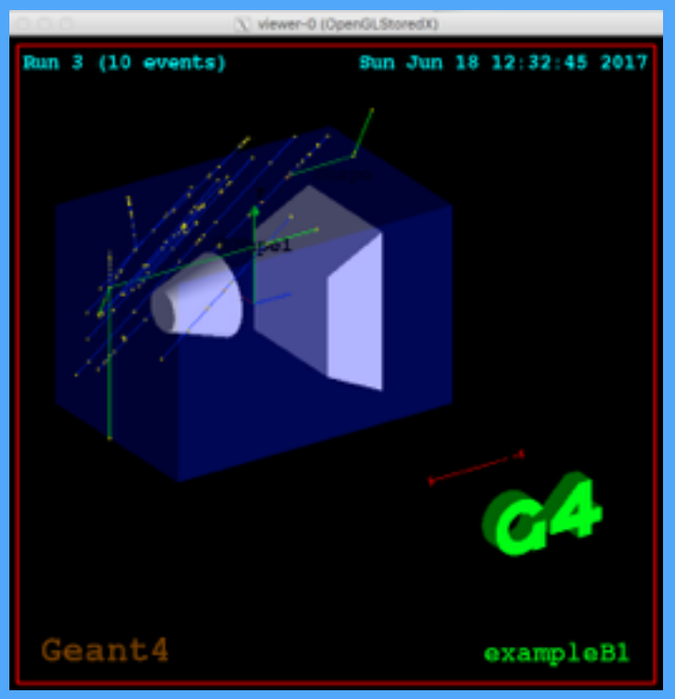
The user interface - like visualisation - is great for setting up and exploring. When want to run seriously you need to get the options set to what you want automatically. Can be done by .mac files or (better(?)) in the code.

How the interface works..

Question: why can you change the direction of the particles in the standard B1 example but not their position?







Normal

```
gun/position 1 2 3 cm    gun/direction 0 .6 .8
```

Answer

Direction is set up in the constructor (in the standard exampleB1)

It then gets over-written by the interface

Position is set up in GeneratePrimaries, over-writing whatever is set up in the interface

Advice for a specific application: put as much as possible in GeneratePrimaries

Here's a more specific example.

Adapt to taste

```
void B1PrimaryGeneratorAction::GeneratePrimaries(G4Event* anEvent)
{
  // start in 1 cm xy square at z=-100
  G4double size = 1.0*cm;
  G4double x0 = size *  (G4UniformRand()-0.5);
  G4double y0 = size *  (G4UniformRand()-0.5);
  G4double z0 = -10.0*cm;
  fParticleGun->SetParticlePosition(G4ThreeVector(x0,y0,z0 ));

  // isotropic distribution
  G4double phi=2*M_PI*G4UniformRand();
  G4double cost=2.0*(G4UniformRand())-1;
  G4double sint=sqrt((1.-cost)*(1.+cost));
  fParticleGun->SetParticleMomentumDirection(G4ThreeVector(sint*cos(phi),sint*sin(phi),cost ));

  // Gaussian energy distribution, mean 100 MeV, sigma 5 MeV
  fParticleGun->SetParticleEnergy(G4RandGauss::shoot(100.0,5.0)*MeV);

  fParticleGun->GeneratePrimaryVertex(anEvent);
}
```

Source: alternative to gun

G4GeneralParticleSource alternative to G4ParticleGun

Much richer User Interface - you can specify distributions in position, direction, energy as a command (or in the .mac file) rather than in the code.

1) Replace all Gun stuff by Source equivalent, deleting lots. Just creation and deletion and a call to fParticleSourcce -> GeneratePrimaryVertex in Generate Primaries
2) Run program, type help, explore gps(option 14). Can generate distributions (uniform, normal, …) for position, direction and energy. And lots more complicated stuff like multiple sources. Enter by hand or put into .mac file

The Scoring Manager

Alternative to that Stepping/Event/RunAction stuff

Add the header file G4ScoringManager.hh to your main program, and add the line

```
G4ScoringManager* s = G4ScoringManager::GetScoringManager();
```

When you compile you get an ‘unused variable’ warning
(which you can avoid by **`s -> setverbose(1);`** or by omitting the assignment)

Run the executable - when you type ‘help’ you have an extra entry in the list

Now you need to specify

1. Mesh
2. Scoring quantity
3. Filter (optional)
4. Output method

The Mesh

Geometry is independent of Physical/Logical volumes

You need to specify
The type - box or cylinder
`/score/create/boxMesh myMeshName`
`/score/create/cylinderMesh myMeshName`

The size - it uses half-sizes like boxes.
`/score/mesh/boxSize and /Cylindersize`
The number of bins `/score/mesh/nBin`
The origin and rotation (default is 0,0,0 and unrotated)

You can have several meshes but we'll just use one
`/score/list` is useful

Example in **`examples/advanced/runAndEvent/RE03`**

The Scorer

What quantity to score? Many available

energyDeposit doseDeposit trackLength cellFlux…
See help for list
`/score/quantity/energyDeposit Edep MeV`
Name is your choice. Units optional.

You can define your own scorer but (i) that's complicated and (ii) what you want is probably provided anyway

You can have several scorers for the same mesh

When done with definitions say `/score/close`

Filters

Come after scorer created but before close (or next scorer)

charged or neutral

KineticEnergy (low and high bounds)

particle (one or more particles)

particleWithKineticEnergy (combines both)

They get given a name

```
/score/filter/particle myFilter proton
```

File Output

After the particles have all been processed you can dump to file

- `/score/dumpQuantityToFile meshname scorername filename`

- `/core/dumpAllQuantitiesToFile meshname filename`

CSV format 6 entries per line: i,j,k bin numbers, mean value, mean squared value and number of entries

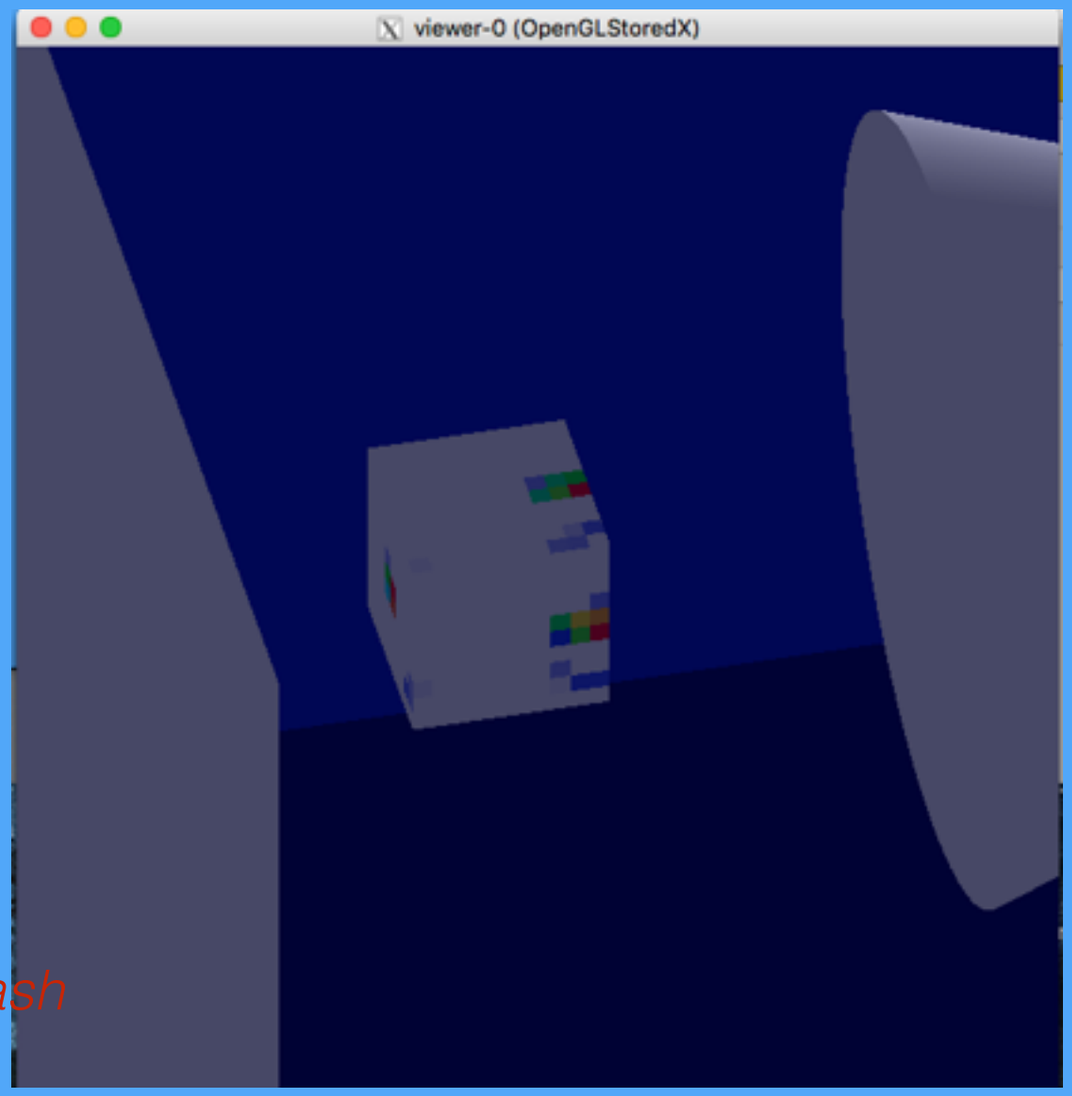
Example

```
/run/initialize

/gps/ene/type Mono
/gps/ene/mono 100 MeV
/gps/particle e-

/score/create/boxMesh myBox
/score/mesh/boxSize  1. 1. 1. cm
/score/mesh/translate/xyz 0. 0. 5. mm
/score/mesh/nBin 10 10 10
/score/quantity/energyDeposit eDep  MeV
/score/quantity/cellFlux Flux
/score/close
/score/list

/run/beamOn 10

/score/dumpQuantityToFile myBox eDep file.txt
/vis/scene/create newscene
/vis/sceneHandler/attach  newscene
/vis/drawVolume world
/score/drawProjection myBox Flux
/vis/drawView 80 40
/vis/viewer/zoom 4
```

This stuff needed to avoid program crash

Part of output file

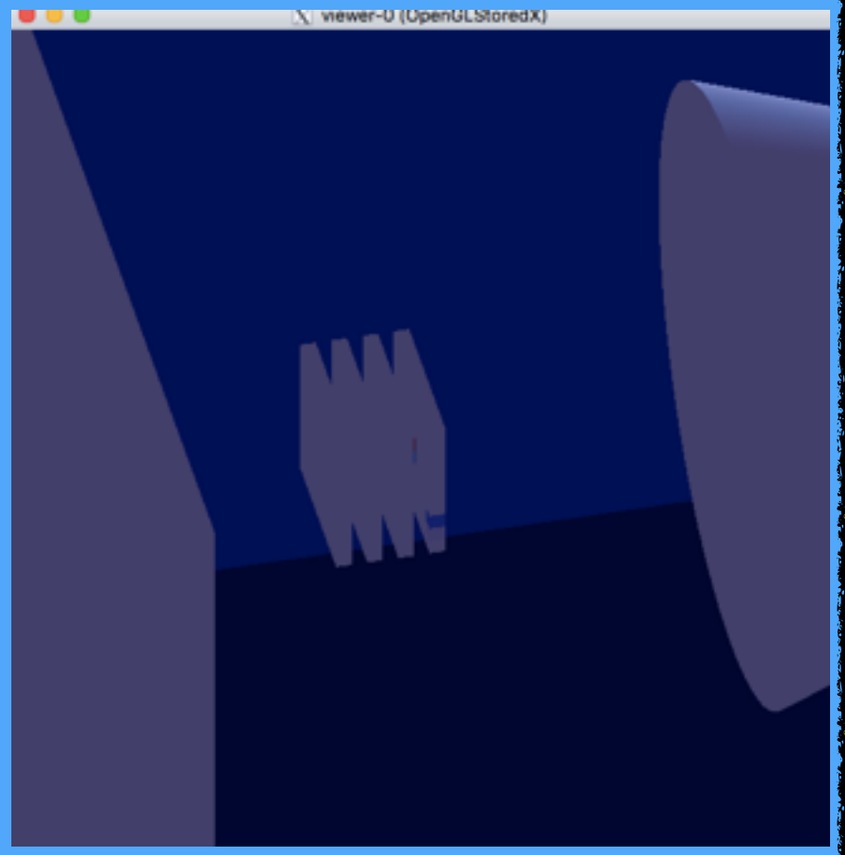
```
4,3,8,0,0,0
4,3,9,0,0,0
4,4,0,0.4104801148281674,0.1684939246693455,2
4,4,1,0.4784187983481977,0.2288845466129334,2
4,4,2,0.002969026344251523,8.815015083564476e-06,3
4,4,3,0,0,0
4,4,4,0,0,0
```



Other options

Plot slices by /score/drawColumn

```
/vis/sceneHandler/attach newScen
/vis/drawVolume world
/score/drawColumn myBox Flux 0 2
/score/drawColumn myBox Flux 0 4
/score/drawColumn myBox Flux 0 6
/score/drawColumn myBox Flux 0 8
```



Cylindrical meshes

see Asai's SLAC slides for moore

Assignment

Demonstrate the Bragg peak for proton energy loss in water using a cylindrical scoring mesh and a general particle source, with beam energies of 200 MeV (Aster), 300 MeV (Dario) and 400 MeV (Mert)

What is the difference between the energy loss and the dose?

Find the radius around the beam enclosing 90% of the dose