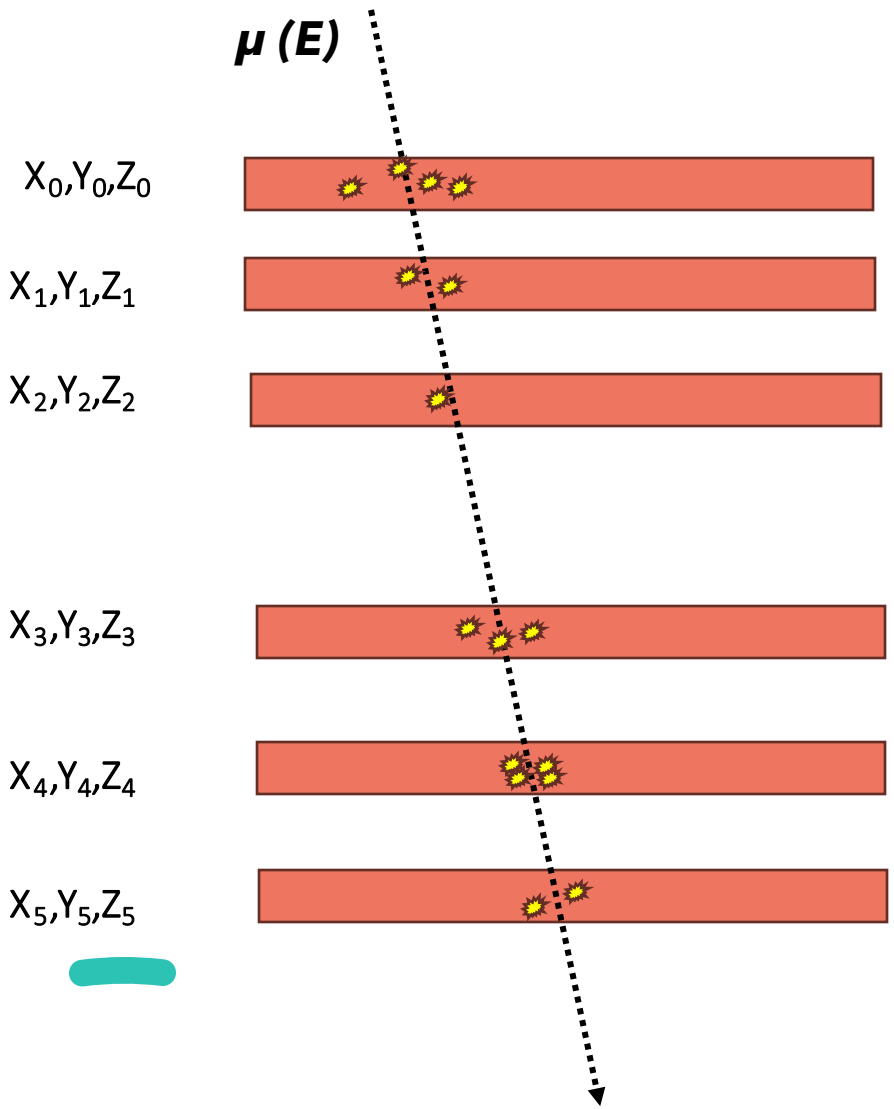
Input data for reconstruction

Next Step:



- Track **only muons** (we **can do** it **in MySensitiveDetector.cc** when we insert the hit)

```
if(abs(track->GetDefinition()->GetPDGEncoding()) == 13)
  {
    G4VPhysicalVolume* volume
    = aStep->GetPreStepPoint()->GetTouchableHandle()->GetVolume();

    G4int PannelCopyNo = volume->GetCopyNo();

    auto hit = new PannelHit();

    hit->SetPos(prestep->GetPosition())
    hit->SetEdep(edep);
    hit->SetPDG(pdg);
    hit->SetPannelID(PannelCopyNo);
    hit->Print();
    fHitsCollection->insert(hit);

}
    return true;
}
```

- Check if **muons cross** the whole detector=>**Pannel: 0,1,2,3,4,5**
- Since we can have multiple hit for the same event =>**calculate the average X,Y,Z position in each plane**
- **Save** the position information and μ initial **energy in** a **.csv file**
- **Write an external script**

- **Upload** makeclass.C and ExternalFct.h
- **Place it** :../micromamba/envs/geant-root/share/Geant4-11.0.3/examples/basic/B1_BNDSchool1/build

Create input data for reconstruction algorithm

- Use makeclass.cc to create an class related to your root output (Tracking.c and Tracking.h):

```
root –l makeclass.C
```

- Open your Tracking.C include the external fct:

```
include "ExternalFct.h"
```

- In Tracking::Loop() :Specify number of plane :

```
int Nplane = 6;
```

- Create your output file '.csv' that should contain : Event, X0,Y0,Z0,X1,Y1,Z1...X5,Y5,Z5,E_generator:

```
ofstream myfile;
    std::string filename = "Detector.csv";
    myfile.open(filename);
//first ligne
    myfile << "Event" << ",";
    for(int i =0;i<Nplane;i++){
      myfile << "X"+std::to_string(i)<<","
      << "Y"+std::to_string(i)<<","
      << "Z"+std::to_string(i)<<",";
    }
    myfile << "E"
    << endl;
```

Create input data for reconstruction algorithm

- Go to the loop over nentries and start your analysis:
  - Loop over NGenPart:

```
 for (int t=0; t<NGenPart;t++){
}
```

  - In the generator loop check if your particle cross whole planes using MuonCrossPlanes method

```
for (int t=0; t<NGenPart;t++){
 if (MuonCrossPlanes(NPannelHit, Nplane, HitPannelID)){
```

  - In your if condition
    - Use GetHit methods to have the average position of muons in each plane:

```
std::vector<std::vector<Double_t>> Hits = GetHits(Nplane,NPannelHit,HitPannelID,HitPosX,HitPosY,HitPosZ);
        std::vector<Double_t> X,Y,Z ;
        X = Hits[0];
        Y = Hits[1];
        Z = Hits[2];
```

    - Fill your output file : Event, X,Y,Z for each plane and GenPartE[t]:

Create input data for reconstruction algorithm

```
myfile << Event << ",";
        for(int i =0;i<Nplane;i++){
          myfile << X[i] <<","
          << Y[i] <<","
          << Z[i] <<",";
        }
        myfile << GenPartE[t]
        << endl;
  } //coincidence condition
} //Loop over particle gene
} // nentries loop
```

- Close your file:

```
  myfile.close();
}//void fct
```

- Compile the code :

```
root –l Tracking.C
Tracking t
t.Loop()
```