

Dec 28, 14 11:07

TrackerSD.cc

Page 1/4

```
//
// *****
// * License and Disclaimer *
// *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// *
// * Neither the authors of this software system, nor their employing *
// * institutes, nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *
// *****
//

#include "TrackerSD.hh"
#include "TrackerHit.hh"
#include "G4VPhysicalVolume.hh"
#include "G4LogicalVolume.hh"
#include "G4Track.hh"
#include "G4Step.hh"
#include "G4ParticleDefinition.hh"
#include "G4Touchable.hh"
#include "G4TouchableHistory.hh"
#include "G4SystemOfUnits.hh"
#include "G4ios.hh"

TrackerSD::TrackerSD(G4String name)
: G4VSensitiveDetector(name)
{
    collectionName.insert("trackerHitCollection");
}

TrackerSD::~TrackerSD()
{
}

void TrackerSD::Initialize(G4HCofThisEvent*)
{
    TkHitCollection = new TrackerHitsCollection(SensitiveDetectorName, collectionName[0]);
    verboseLevel = 0;
    HitMap.clear();
    detId=0;
    trackID=0;
}

G4bool TrackerSD::ProcessHits(G4Step*aStep, G4TouchableHistory*){

    if(verboseLevel>1)
        std::cout<< " Entering a new Step " << aStep->GetTotalEnergyDeposit() << " given by Track " << aStep->GetTrack()->GetTrackID() << " with charge " << aStep->GetTrack()->GetDefinition()->GetPDGCharge() << " in the volume " << aStep->GetPreStepPoint()->GetPhysicalVolume()->GetLogicalVolume()->GetName() << std::endl;

    if(aStep->GetTotalEnergyDeposit()>0. && 0.0 != aStep->GetTrack()->GetDefinition()->GetPDGCharge()){
```

Dec 28, 14 11:07

TrackerSD.cc

Page 2/4

```
    if(verboseLevel>1)
        std::cout<< " I'm going to check if I need a new Hit or I have to update the old one " << std::endl;
    if(NewHit(aStep)){
        CreateHit(aStep);
    }else{
        UpdateHit(aStep);
    }
    return true;
}

G4bool TrackerSD::NewHit(G4Step* aStep){

    if(verboseLevel>1)
        std::cout<< " I'm inside the NewHit method " << std::endl;
    G4Track * theTrack = aStep->GetTrack();
    if(verboseLevel>1){
        std::cout<< " new TrackId = " << theTrack->GetTrackID() << " old TrackId = " << trackID << std::endl;
        std::cout<< " new DetId = " << SetDetectorId(aStep) << " old detId = " << detId << std::endl;
    }
    if(theTrack->GetTrackID()!=trackID || SetDetectorId(aStep)!=detId){
        return true;
    }
    return false;
}

int TrackerSD::SetDetectorId(G4Step* aStep){
    int detId = 0;
    if(verboseLevel>1)
        std::cout<< " I'm inside the SetDetectorId method " << std::endl;
    const G4VTouchable* VT(aStep->GetPreStepPoint()->GetTouchable());
    if(verboseLevel>1)
        std::cout<< " Volume Name Layer? = " << VT->GetVolume(4)->GetName() << " and CopyNumber = " << VT->GetCopyNumber(4) << std::endl;
    if(VT->GetCopyNumber(4)==0)
        detId=2200;
    else
        detId=2100;
    if(verboseLevel>1)
        std::cout<< " Volume Name Ladder? = " << VT->GetVolume(2)->GetName() << " and CopyNumber = " << VT->GetCopyNumber(2) << std::endl;
    if(VT->GetCopyNumber(2) == 0)
        detId+=30;
    else if (VT->GetCopyNumber(2) == 1)
        detId+=20;
    else if (VT->GetCopyNumber(2) == 2)
        detId+=10;
    if(verboseLevel>1)
        std::cout<< " Volume Name Module? = " << VT->GetVolume()->GetName() << std::endl;
    if(VT->GetVolume()->GetName()=="SiliconSensorM")
        detId+=1;
    else
        detId+=2;
    if(verboseLevel>1)
        std::cout<< " DetId = " << detId << std::endl;
    return detId;
}

void TrackerSD::CreateHit(G4Step * aStep){
    G4Track * theTrack = aStep->GetTrack();
    if(verboseLevel>1)
        std::cout<< "TrackerSD::CreateHit Start to collect the info for the new Hit " << std::endl;
    int theDetectorId = SetDetectorId(aStep);
    if (theDetectorId == 0){
        std::cout<< " Error: theDetectorId is not valid. ";
    }
    int theTrackID = theTrack->GetTrackID();
```

Dec 28, 14 11:07

TrackerSD.cc

Page 3/4

```

G4double theEnergyLoss      = aStep->GetTotalEnergyDeposit()/MeV;
G4ThreeVector theExitPoint = aStep->GetPostStepPoint()->GetPosition();
G4ThreeVector theEntryPoint = aStep->GetPreStepPoint()->GetPosition();
G4double thePabs           = aStep->GetPreStepPoint()->GetMomentum().mag()/MeV;
G4double theTof            = aStep->GetPreStepPoint()->GetGlobalTime()/nanosecond;
G4int theParticleType      = theTrack->GetParticleDefinition()->GetPDGEncoding();

G4ThreeVector gmd = aStep->GetPreStepPoint()->GetMomentumDirection();
// convert it to local frame
G4ThreeVector lmd = ((G4TouchableHistory *) (aStep->GetPreStepPoint()->GetTouch
able()))->GetHistory()->GetTopTransform().TransformAxis(gmd);
G4double theThetaAtEntry = lmd.theta();
G4double thePhiAtEntry = lmd.phi();

if(verboseLevel>1)
    std::cout<<"TrackerSD::CreateHit I'm creating the new Hit on DetId "<<theDetectorId<<std::endl;

detId=theDetectorId;
trackID=theTrackID;
TrackerHit* trackerHit = new TrackerHit(theEntryPoint,theExitPoint,thePabs,the
Tof,
                                theEnergyLoss,theParticleType,theDetec
torId,
                                theTrackID,theThetaAtEntry,
                                thePhiAtEntry);

G4int cell = TkHitCollection->insert(trackerHit);
int mapKey = ((trackID&tkIdMask)<<tkIdOffset)|(detId&detIdMask);
HitMap[mapKey] = cell-1;
if(verboseLevel>1)
    std::cout<<"TrackerSD::CreateHit I have just created the new Hit"<<std::endl;
}

void TrackerSD::UpdateHit(G4Step* aStep){
    if(verboseLevel>1)
        std::cout<<"TrackerSD::CreateHit I just going to update the Hit"<<std::endl;
    int mapKey = ((aStep->GetTrack()->GetTrackID())&tkIdMask)<<tkIdOffset)|(SetDete
ctorId(aStep)&detIdMask);
    if(HitMap.find(mapKey)!=HitMap.end()){
        G4double theEnergyLoss      = aStep->GetTotalEnergyDeposit()/MeV;
        (*TkHitCollection)[HitMap[mapKey]]->SetExitPoint(aStep->GetPreStepPoint()->G
etPosition());
        (*TkHitCollection)[HitMap[mapKey]]->AddEnergyLoss(theEnergyLoss);
        if(verboseLevel>1)
            std::cout<<"TrackerSD::CreateHit I just have update the Hit"<<std::endl;
    }
}

void TrackerSD::EndOfEvent(G4HCofThisEvent* HCE)
{
    static G4int HCID = -1;

    HCID = GetCollectionID(0);
    HCE->AddHitsCollection( HCID, TkHitCollection );
}

void TrackerSD::clear()
{
}

void TrackerSD::DrawAll()
{
}

void TrackerSD::PrintAll()
{
}

```

Dec 28, 14 11:07

TrackerSD.cc

Page 4/4

}