Project for the GEANT4 Tutorial, September 2018

Eva Brottmann Hansen, Lund University, eva.hansen@hep.lu.se

October 18, 2018

In this report the results of simulating with GEANT4 the scanning of a container with different types of beams are presented. The container is made of aluminium and has 5 mm thick walls. Inside is a person, three boxes full of water melons, and a ball of uranium which is shielded by a beautiful marble table. The container is placed beside a CsI detector for measuring neutrons and gammas, and above and below it, scintillators for detecting muons. Figure 1 shows an image of the simulated detector with the contents and detectors.

A neutron beam of 15 MeV and a gamma beam of 5 MeV are used to scan in the direction perpendicular to the side walls. They are "fired" using particle gun with the vertex position randomized in a plane on the opposite side of the container from the neutron/gamma detector. A few example events is shown for the neutron and gamma beam in Figure 2 and Figure 3, respectively.

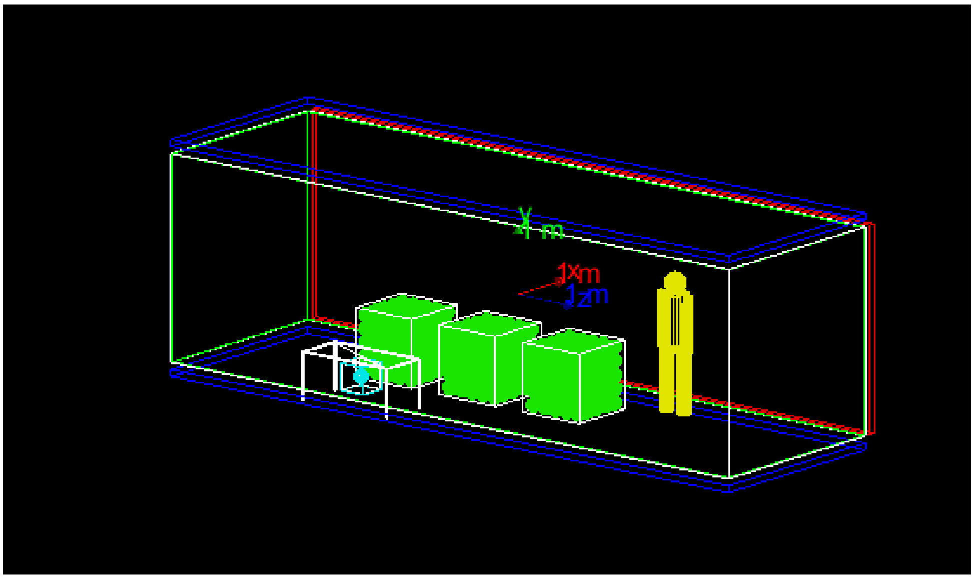

Figure 1: Image displaying the contents of the truck.

The energy deposit in the CsI detector is recorded using command-based scoring. In Figure 4 and 5 shows that the contours of the person and the three boxes is seen clearly by the energy deposit by scanning with either gamma or neutron beam. The uranium ball, hiding behind one the boxes, can also be spotted as a white spot in the box to the right in the picture.

For scanning in the top-to-bottom direction, the abundant source of cosmic ray muons can be used. For each muon, one hit position is recorded in the top scintillator and one in the bottom detector in order to calculate the muons scattering angle. For this purpose, some new classes are implemented, which inherit from the classes G4SensitiveDetector, G4HitClass, G4RunAction, and G4EventAction. From Figure 6 we see that this method does not nearly show the objects in the

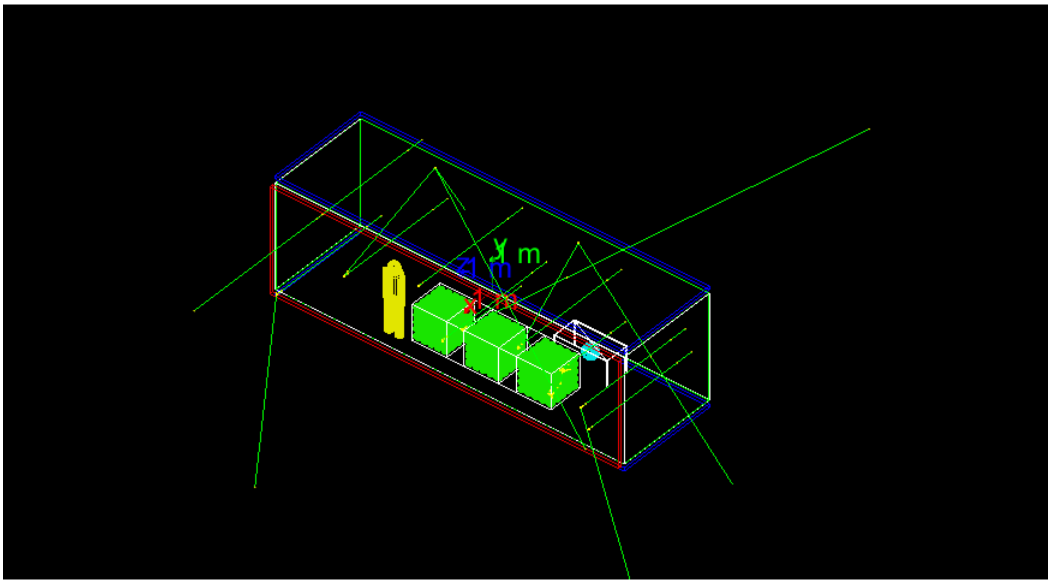

Figure 2: Example showing 10 gamma events

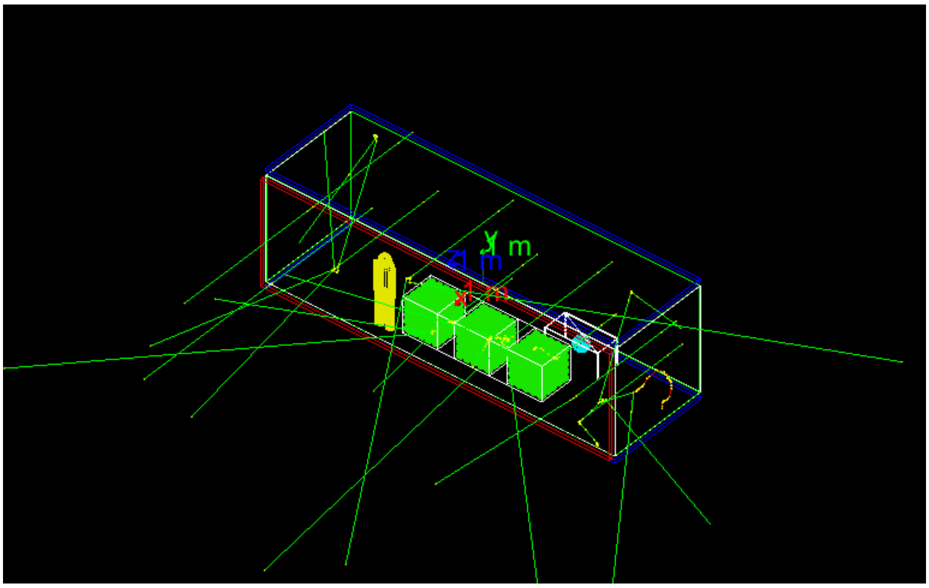

Figure 3: Example showing 10 neutron events.

container to the same precision as the neutron/gamma scan. The uranium ball appears around $(x, z) = -1, -2$ m and the person is seen vaguely at $(x, z) = 1, 2$ m, but the boxes of watermelons are not visible.

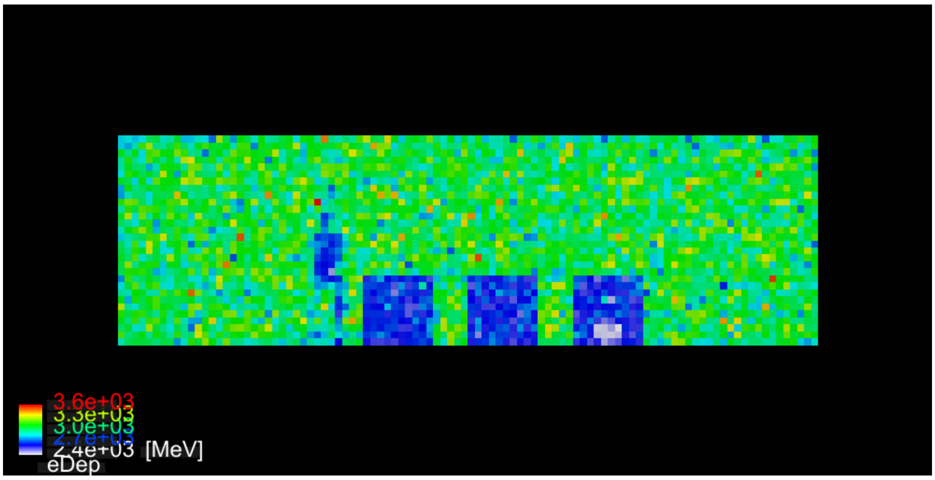

Figure 4: Energy deposit by 10^5 gammas.

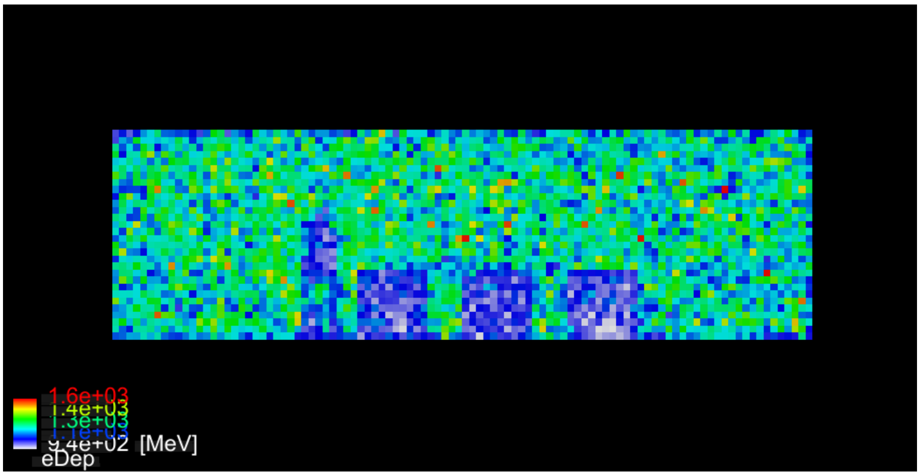

Figure 5: Energy deposited in the CsI detector by 10^5 neutrons.

1 Main function

[h!]

```
#include "DetectorConstruction.hh"
#include "ActionInitialization.hh"

#ifdef G4MULTITHREADED
#include "G4MTRunManager.hh"
#else
#include "G4RunManager.hh"
#endif

#include "G4ScoringManager.hh"
#include "G4UImanager.hh"
#include "FTFP_BERT.hh"
#include "G4StepLimiterPhysics.hh"

#include "G4VisExecutive.hh"
#include "G4UIExecutive.hh"
```

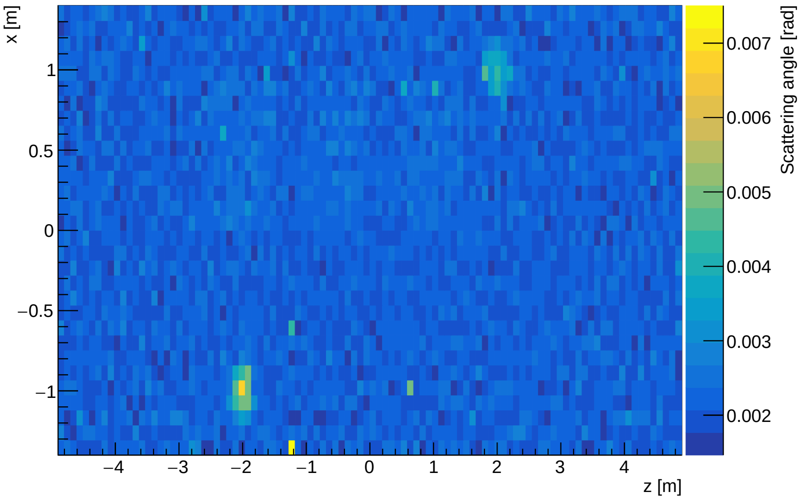

Figure 6: Image displaying the contents of the truck.

```
//....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo........oooOO0OOooo......

int main(int argc,char** argv)
{
  //Detect interactive mode (if no argument) and define UI session
  G4UIExecutive* ui = 0;
  if ( argc == 1 ) { //No commands line argument
    //Let G4UIExecutive guess what is the best available UI
    ui = new G4UIExecutive(argc,argv);
  }

  // Construct the default run manager
  // Note that if we have built G4 with support for Multi-threading we set it here
#ifdef G4MULTITHREADED
  G4MTRunManager* runManager = new G4MTRunManager;

    //Set the default number of threads to be the number of available cores of the
        machine
    //If not specified use 2 threads
    //runManager->SetNumberOfThreads( G4Threading::G4GetNumberOfCores() );
#else
  G4RunManager* runManager = new G4RunManager;
#endif

  // Activate UI-command base scorer
    G4ScoringManager * scManager = G4ScoringManager::GetScoringManager();
    scManager->SetVerboseLevel(1);

    // Mandatory user initialization classes

    //====================
    //The Geometry
    runManager->SetUserInitialization(new DetectorConstruction);
    //====================
    //The Physics
    G4VModularPhysicsList* physicsList = new FTFP_BERT;
    physicsList->RegisterPhysics(new G4StepLimiterPhysics());

    runManager->SetUserInitialization(physicsList);

    //====================
    // User action initialization
    runManager->SetUserInitialization(new ActionInitialization());

    // Visualization manager construction
    G4VisManager* visManager = new G4VisExecutive;
    visManager->Initialize();

    // Get the pointer to the User Interface manager
    G4UImanager* UImanager = G4UImanager::GetUIpointer();

    if (argc>1) {
        // execute an argument macro file if exist
        G4String command = "/control/execute ";
        G4String fileName = argv[1];
        UImanager->ApplyCommand(command+fileName);
    }
    else {
      //We have visualization, initialize defaults: look in init_vis.mac macro
      UImanager->ApplyCommand("/control/execute init_vis.mac");
      if ( ui->IsGUI() ) {
         UImanager->ApplyCommand("/control/execute gui.mac");
      }
      ui->SessionStart();
      delete ui;
    }
    // Job termination
    // Free the store: user actions, physics_list and detector_description are
    // owned and deleted by the run manager, so they should not be deleted
    // in the main() program !

    delete visManager;
    delete runManager;
    //delete UImanager;

    return 0;
}
```