Overview

For this assignment, continue building on your code from Assignment 6. Now that you have a simple geometry and particle source set up, and the ability to record what happens, it's time to start making things dynamic and configurable.

You will take advantage of the messenger classes we went over in lecture and use them to run your simulation in different ways without needing to recompile the code in between runs.

You'll want to update your code to include the RunMessenger we put together in class. This will allow you to change the file path for the Hits file, saving you the trouble of renaming it between runs (and preventing associated mistakes!). The code can be found here: https://github.com/micahfolsom/g4-ne697

In this assignment, you'll select on energy depositions by electrons ("e-") to get a clearer radiograph, since this will filter out events where, for example, the only Hit is the gamma stepping out of the detector.

To complete this assignment: make the changes listed in the *Coding Tasks* section, and test them as you implement them. Once you're satisfied that they work, execute the simulations described in the *Run* section, plot the results, and save the images to the **images/** directory.

Code Submission

Create a folder in your Github repository named **assignment7/**.

For each part of the questions below (where applicable), create a separate .mac file to accomplish the task, just like in previous assignments. This is especially important now that we are modifying the geometry - we can only do this before /run/initialize, so different geometries will need to be run from different .mac files.

Place your .mac files in the **scripts/** sub-folder so they get copied to the build directory when you run CMake. You may optionally update your CMakeLists.txt like we did in class so it copies every time you type *make*, which I've found to be helpful.

This assignment will be due by **Tuesday, November 30th at midnight Pacific Time.**

Geometry and Source

Your T and parallel beam source will remain mostly the same. You will be modifying the radiation detector and other aspects of the program with your messenger classes.

Coding Tasks

First, you will implement the changes in this section. Then you'll run the code with the parameters in the next section, saving the radiographs you produce in an images/ directory.

1. Add a GeometryMessenger like we covered in class, attached to the DetectorConstruction. It should have the following commands to manipulate the geometry:
    a. **/ne697/geometry/det_thickness <double> <unit>**
        i. The full thickness (Y-dim) of the detector panel. Must be positive.
        ii. If the user provides a negative value, print some feedback to the error output stream (**G4cerr** instead of **G4cout**), and don't apply the command.
        iii. Default: 5 cm
    b. **/ne697/geometry/det_size <double> <unit>**
        i. The full X and Z extents of the detector panel. It's always square.
        ii. Default: 50 cm
2. Add a new messenger called MaterialMessenger. This will be very similar to the GeometryMessenger, and it should also be attached to the DetectorConstruction (so it will just have 2 messengers. No problem here). This will have one command to change the detector material to a G4NistManager pre-built:
    a. **/ne697/material/det_material <string**>
    b. For example, to change the material to "G4_Ge" (germanium
        i. /ne697/material/det_material G4_Ge
        ii. There's technically a way to check if it's a valid material but don't worry about it for this assignment
        iii. Default: G4_SODIUM_IDODIDE
3. Add a new messenger called GunMessenger. You will use this to modify the particle source.
    a. **/ne697/gun/offset <double> <unit>**
        i. Distance of the plane source from the T. Just like the detector offset, this should always be a positive value, then you figure out what the actual coordinate is.
        ii. If the user provides a negative value, print some feedback to the error output stream (**G4cerr** instead of **G4cout**), and don't apply the command.
        iii. Default: 30 cm
4. Add some logic in your SensitiveDetector class, in the ProcessHits() function, to only generate Hits if:
    a. The particle is "e-"
    b. Energy deposited > 0

When implementing these features, I recommend doing one at a time, and testing with the visualization to confirm that your command took hold. For the geometry and materials commands, make sure you're entering them **before** /run/initialize is called! I also recommend committing with each successfully-implemented command, so if something breaks, it'll be easier to figure out where.

Don't worry too much about range-checking the values, such as making sure the detector isn't bigger than the world, except where noted. While this is nice to have, I would like you to focus on getting the basic functionality implemented. You can always improve it when you eventually copy/paste it into another simulation!

Runs

With all of these things made dynamic, run the following list of simulations to demonstrate the new functionality. For each run, do 100,000 particles. Consider each independently - that is, if one run says to do one set of parameters, assume the next run is back to using everything as defaults, except for the items specified.

- run1.mac
    - Run with the defaults. This should be 1 MeV gamma rays moving in the -Y direction from Y=30 cm, with random positions in the XZ plane from -25 to +25 cm
- run2.mac
    - Make the detector 1 cm thick
- run3.mac
    - Change the gun energy to 300 keV
- run4.mac
    - Change the gun offset to 50 cm
- run5.mac
    - Change the detector material to G4_Ge
    - Use a gun energy of 300 keV
- run6.mac
    - Collect gamma hits instead of e- hits (you will need to change the code and recompile)

Radiographs

To create the radiographs, take the position of the first Hit from each event ID. The e- hits, compared to the last assignment, should have much better contrast since now (by default) you're selecting only energy depositions, and we're ignoring the gammas entirely.

Place the corresponding images from runs 1-6 into the **images/** directory in your repository.