11-13 January 2021

DAMICM software School

DAMICM G4 simulations

Claudia De Dominicis
Mariangela Settimo

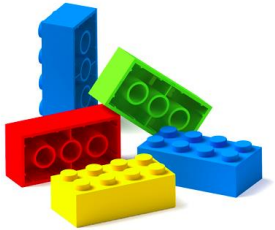
Overview

- Geant4 Introduction
- DAMICM G4 simulations code:
  - Primary particles
  - Physics List
  - Geometry
  - Analysis
- Useful Links
- Exercises

N.B. This talk provides ONLY the basics to use G4 and the DAMICM code, this is not a complete description!

GEANT4

General purpose Monte Carlo **toolkit** for simulating the passage of elementary particles through and interacting with matter



Geant4 provides building blocks (bricks)

Users have to assemble them to describe their scenario in their application program

G4 Application

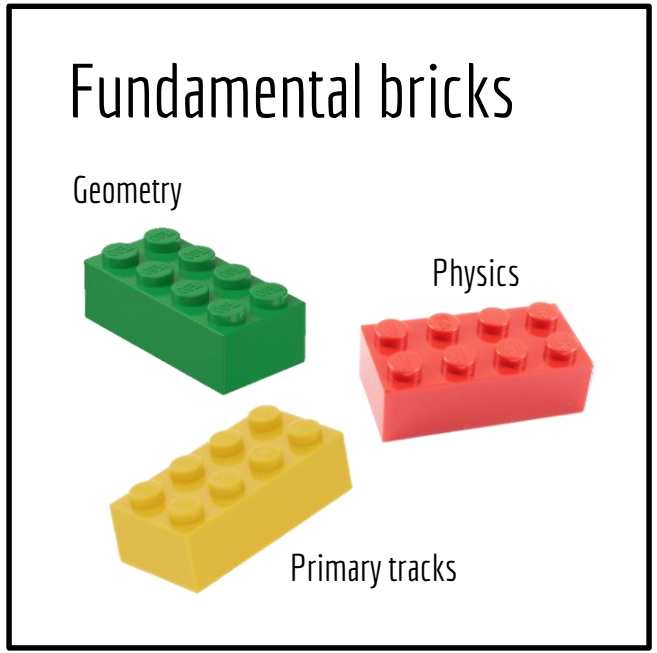
You have to define:

- Your geometrical setup (materials, volumes)
- Involved physics (particles, physicsprocesses/models, production thresholds)
- How an event starts (primary track generation)

You may also want:
To extract useful information, use (Graphical) User Interface, perform optional user actions

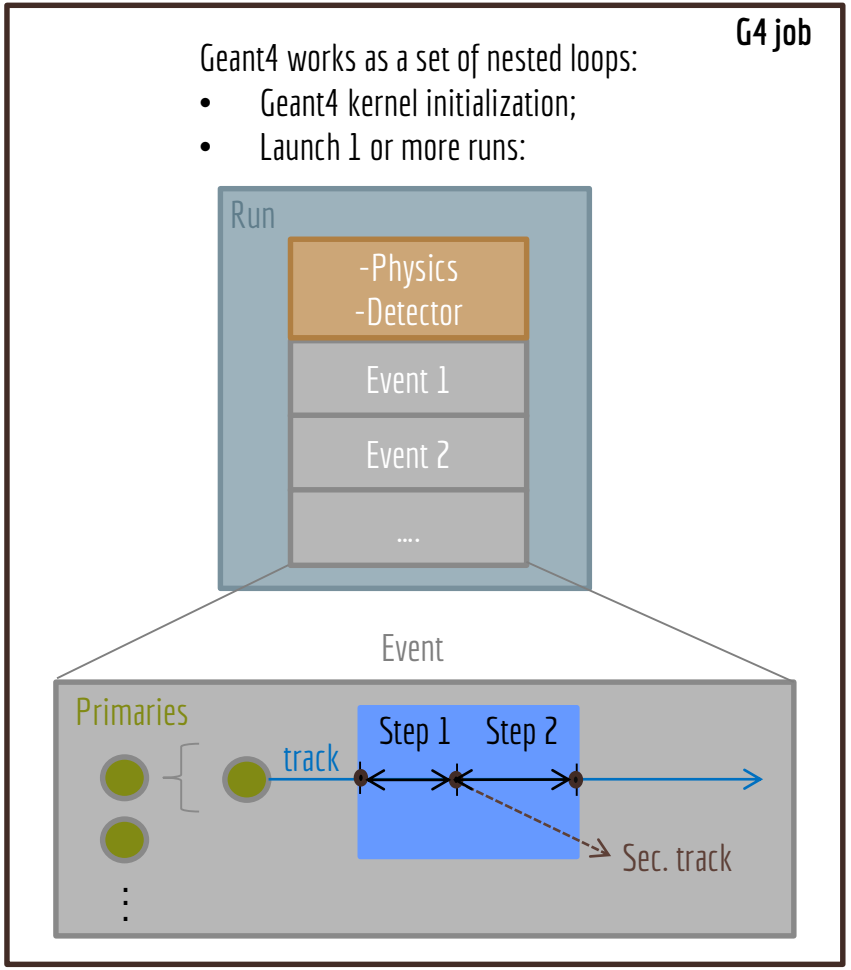
This is done by means of:

- main( ) program: c++ function
- User initialization classes (mandatory)
  G4VUserDetectorConstruction, G4VPrimaryGeneratorAction, G4VUserPhysicsList
- User action classes (optional)
  G4UserRunActionEvent, G4UserEventAction , G4UserTrackingAction, G4UserSteppingAction, G4UserStackingAction



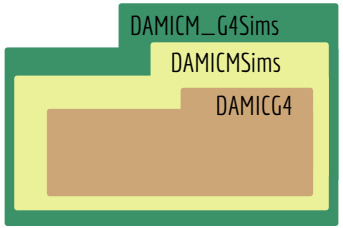
G4 keywords

- Particle properties:
  Mass, pdg ID, charge, momentum, position
- Track properties:
  - Track ID (=1 if primary particle)
  - G4 track class
- Step properties:
  - Step ID
  - G4 step class



DAMICM G4 simulation code

- A DAMICM G4 simulation code has been developed to be used for DAMIC-M simulations. The code is developed in Geant4 10.4.



- Code available on git lab

- Versions:
  - release: **v1.1.0** ⟵ Used in this school!!!
  - development: e.g. shielding, version for G4 10.6

- Validation: compare to reference plots in case of major changes in the code or new G4 versions

Structure of DAMICM G4 simulation code

DAMICG4

- **DAMICG4.cc**:

  main( ) function : initialization Detector construction, Physics list, define Primary generator class

src*

- **DAMICPhysicsListLivermore.cc**: Definition of physics processes, of energy range and production cuts for particles
- **DAMICMPrimaryGeneratorAction.cc**

  DAMICParticleSource.cc : generation of primary particles with different spatial, momentum, energy distributions

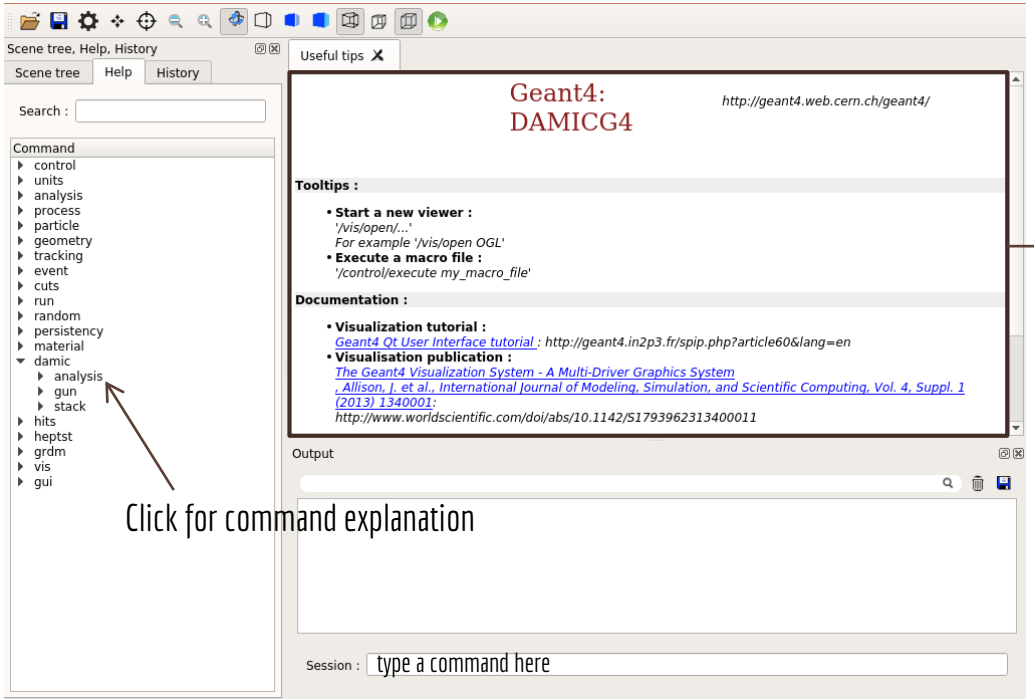
  DAMICParticleSourceMessenger.cc: definition of user commands
- **DAMICDetectorCostruction.cc**: builds detector by reading from GDML file
- **DAMICActionInitialization.cc** : Instantiates all initialization and action classes called during event processing
- **DAMICAnalysisManager.cc** : creates Ntuples,...

  DAMICAnalysisManagerMessenger.cc : definition of user commands

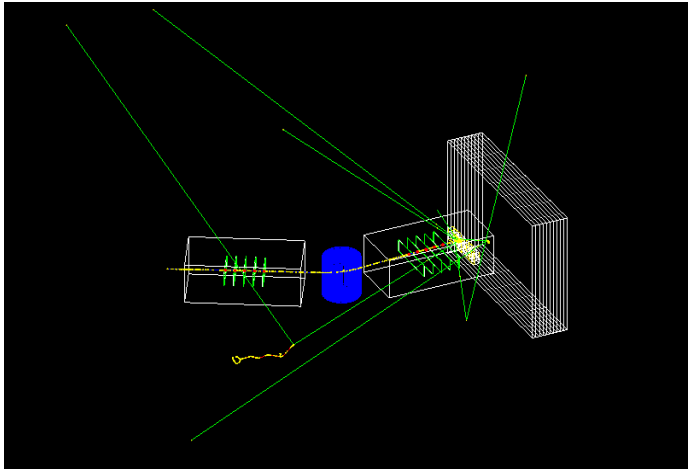
* Corresponding header file in «include» directory

How to run

Interactive mode: ./DAMICG4 gdml



Click for command explanation



Only for few events simulations!!!!!!

How to run

Batch mode: ./DAMICG4 gdml file.mac

Initialize geometry
Physics list

Particle generator

Launch simulation with N evts

file.mac

```
/control/verbose 0
/run/verbose 0
/run/initialize
/tracking/verbose 0
/event/verbose 0
/damic/analysis/setstoreparticleinfo 0
/damic/analysis/setstoretrack 0
/damic/stack/fulldecay 0
/damic/gun/particle ion
/damic/gun/ion Z A 0 0.0
/damic/gun/energy/mono 0 eV
/damic/gun/direction/oned
/damic/gun/direction/onedX 1
/damic/gun/direction/onedY 0
/damic/gun/direction/onedZ 0
/damic/gun/position/dovolume
/damic/gun/position/addvolume LOGICALVOLUME_PV 1 false
/random/setSeeds 23471 6817
/analysis/setFileName file_name
/run/beamOn Nevts
```

mandatory commands

How to run

Save or not trees for debugging informations

Radioactive decay process option (see slide 13)

Batch mode: ./DAMICG4 gdml file.mac

Change seeds to have different simulations

file.mac

```
/control/verbose 0
/run/verbose 0
/run/initialize
/tracking/verbose 0
/event/verbose 0
/damic/analysis/setstoreparticleinfo 0
/damic/analysis/setstoretrack 0
/damic/stack/fulldecay 0
/damic/gun/particle ion
/damic/gun/ion Z A 0 0.0
/damic/gun/energy/mono 0 eV
/damic/gun/direction/oned
/damic/gun/direction/onedX 1
/damic/gun/direction/onedY 0
/damic/gun/direction/onedZ 0
/damic/gun/position/dovolume
/damic/gun/position/addvolume LOGICALVOLUME_PV 1 false
/random/setSeeds 23471 6817
/analysis/setFileName file_name
/run/beamOn Nevts
```

Particle generator

DAMICParticleSource.cc

- Definition of primary particles, direction distribution, energy distribution
- Communicates with Messenger that communicates with the user through commands

Commands*:

Definition primary particle: **/damic/gun/particle** [particle name] — particle name: e-,mu-, proton,ion etc.
/damic/gun/ion [Z] [A] [Q] [E] — A: atomic mass Z: atomic number, Q: Ion Charge, E: ExitationEnergy[keV]

Directions of primary particle:

One direction: **/damic/gun/direction/oned**
/damic/gun/direction/onedX [value]
Direction following a distribution: **/damic/gun/direction/distri** [distribution] — distribution: ex. Isotropic

(*) you can use also predefined G4 commands

Particle generator

Commands:

Energy of the primary particles:

One energy: **/damic/gun/energy/mono [energy]**
Energy Distribution: **/damic/gun/energy/distri [distribution]**

[distribution]: Uniform

Position of the primary particles:

from a point source: **/damic/gun/position/dosource**
following a shape: **/damic/gun/position/doshape** [shape]
from a specific material: **/damic/gun/position/domaterial** [material]
from different damic geometry volumes**: /damic/gun/position/dovolume**
/damic/gun/position/addvolume [Volume] [W] [M] [min] [max] [s]

[shape]: Para, Sphere, Cyl
[material]: material in GDML, ex. G4_Cu

[Volume]: LogicalVolume_PV (M=false) or LogicalVolume(M=true),
[W]: weight = 1, [M]: Molteplicity (true/false),
[min]/[max]:minimum/maximum copy number,
[s]: step btw min and max

Optional

Only if M is true

M is true when the name of the volume is Volume_CopyNumb_PV

Physics List

DAMICPhysicsListLivermore.cc:

Costruction Particles, implementation Physics Processes, Energy Range, Production cuts

Used physics list: Livermore + radioactive decay + neutrons processes (scattering + fission)

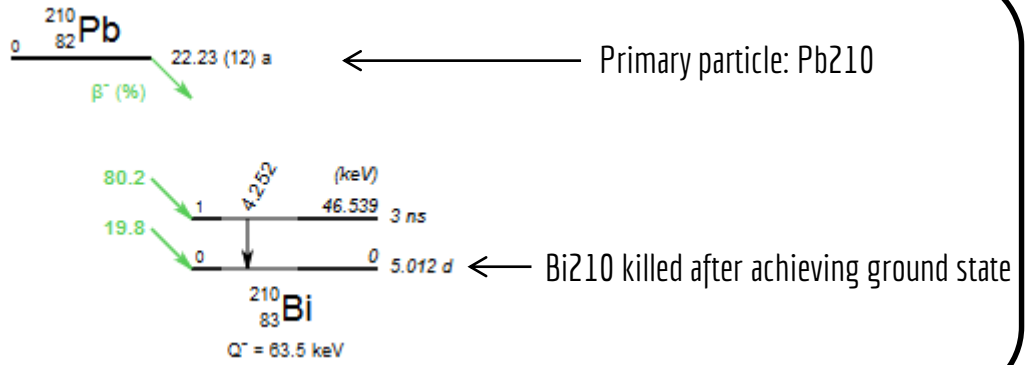
Energy Range: > 20 eV

Radioactive decay

The code by default kills the first son of the decay at ground state

To activate again the full chain simulation:

/damic/stack/fulldecay 1



Production cuts

Definition: Mimum distance a secondary particle has to be able to travel to be produced. Range cut then converted in Energy cut.

- Production cuts for e-, e+, gamma, proton
- To set properly to optimize the simulations
- It is recommended to use a range cut = smallest dimension you're interested in.
- Prod cuts can be set:
  - per particle, Region
  - in PhysicsList, GDML, by command

Commands:
/run/setCut 1mm
/run/setCutForAGivenParticle e- 0.1 mm
/run/setCutForRegion MyRegion 1 mm

Default production cuts:

in Physics list: gamma 0.1 um, e- 1 nm, e+ 0.1 um, proton 0.1 um
in Detector costruction (in case no length is given through Region attribute): 0.00001 mm (*)

(*) For production cuts in Silicon see Collaboration Meeting presentation DOCDB 306-v1

Check production cuts

You can check prod. cuts on terminal when you launch simulation “Table of registered couples”

```
========= Table of registered couples ==============================

Index : 0     used in the geometry : Yes
 Material : G4_Galactic
 Range cuts        :  gamma  100 nm     e-  1 nm     e+  100 nm  proton 100 nm
 Energy thresholds :  gamma  20 eV     e-  20 eV     e+  20 eV  proton 10 eV
 Region(s) which use this couple :
    DefaultRegionForTheWorld
```

You can use /run/dumpCouples
but the code does it by default

Detector Construction

DAMICDetectorCostruction.cc

- Detector Construction from GDML file: It takes everything in the WORLD Volume in the gdml

GDML: The Geometry Description Markup Language is an application-indepedent geometry description format based on XML. It can be used as the primary geometry implementation language as well as it provides a geometry data exchange format for the existing applications.

Detector Construction - GDML

GDML Structure

5 parts:

1. **‹define› … ‹/define› block**: numerical values of different constants, positions etc used later on in the geometry construction.
2. **‹materials› …‹/materials› block**: definitions of all the materials (one element or mixtures).
3. **‹solids› … ‹/solids› block**: collection of all solid definitions (simple or composite)
4. **‹structure› … ‹/structure› block** : adding properties (material,region,prod cuts, sensitive) to solids (logical volumes) and place them in another volume (physical volumes).
5. **‹setup› … ‹/setup› block:** used to specify the top volume of the geometry tree.

```
<?xml version="1.0" encoding="UTF-8"?>
<gdml xsi:noNamespaceSchemaLocation="schema/gdml.xsd">
        <define>
                …
                <position name="TrackerinWorldpos" unit="mm" x="0" y="0" z="100"/>
        </define>
        <materials>
                …
                <element name="Nitrogen" formula="N"  Z="7.">
                <atom value="14.01"/>
                </element>
                <material formula=" " name="Air" >
                        <D value="1.290" unit="mg/cm3"/>
                        <fraction n="0.7" ref="Nitrogen" />
                        <fraction n="0.3" ref="Oxygen" />
                </material>
        </materials>
        <solids>
                …
                <box lunit="mm" name="Tracker" x="50" y="50" z="50"/>
        </solids>
        <structure>
                …
                <volume name="World" >
                        <materialref ref="Air" />
                        <solidref ref="world" />
                        <physvol>
                        <volumeref ref="Tracker" />
                        <positionref ref="TrackerinWorldpos"/>
                        <rotationref ref="TrackerinWorldrot"/>
                        </physvol>
                </volume>
        </structure>
        <setup name="Default" version="1.0" >
                <world ref="World" />
        </setup>
</gdml>
```

Detector Construction - GDML

DEFINE BLOCK

```
<define>

   <variable name="Height_Tube" value="300.0" />
   ...
   <rotation name="norot" unit="degree" x="0" y="0" z="0" />
   <position name="poscenter" unit="mm" x="0" y="0" z="0"/>

</define>
```

MATERIAL BLOCK

```
<materials>

      <element name="G4_Cu" formula="Cu" Z="29.">
            <atom value="63.546"/>
      </element>

      <material name="G4_SILICON_DIOXIDE" formula="SiO_2">
            <D value="2.2" unit="g/cm3"/> <!--density-->
            <fraction n="0.46744" ref="G4_Si"/> <!--fraction of mass-->
            <fraction n="0.53256" ref="G4_O"/>
      </material>

      <material name="Water" formula="H2O">
        <D value="1.0" />
        <composite n="2" ref="Hydrogen" /> <!-- atom count-->
        <composite n="1" ref="Oxygen" />
      </material>

</materials>
```

Detector Construction - GDML

SOLID BLOCK

```
<solids>
        <box lunit="m" name="WorldBox" x="5.0" y="5.0" z="5.0"/>
        <tube name="TubeSolid"  lunit="mm" aunit="degree" z="Height_Tube" rmin="0" rmax="Rmax_Tube" deltaphi="360"/>

        <union name="TubesUnion"> <!-- same thing with “subtraction” instead of “union”-->
                <first  ref=”TubeSolid"/>
                <second ref="TubeSolid"/>
                <position name="pos_Tube" unit="mm" x="0" y="0" z="+Height_Tube"/>
                <!--position center 2nd solid respect to center 1st solid-->
                <rotationref ref="norot"/>
        </union>

        <loop for="i" to="NumberOfReplicas" step="1"> <box name="Box[i]" lunit="mm" x=«Lx" y="Ly" z="Lz"/>

</solids>
```

Detector Construction - GDML

STRUCTURE BLOCK

```
<structure>

  <volume name="TubeVolume">
      <materialref ref="G4_Cu"/>
      <solidref ref="TubeSolid"/>
      <auxiliary auxtype="Color" auxvalue="0.85,0.3,0.3"/>
      <auxiliary auxtype="SensDet" auxvalue="tube"/>
     <auxiliary auxtype="Region" auxvalue="CuRegion, ProdCutValue"/>
  </volume>

  <volume name="TubeStack">
        <materialref ref="G4_Galactic"/>
        <solidref ref="MotherSolid"/>

        <loop for="copyNo" to= " 2" step="1">
                <physvol copynumber="copyNo">
                 <volumeref ref="TubeVolume"/>
                 <position name="Pos" unit="mm" x= " " y= " " z= " "/>
                 <rotation name="Rot" unit="degree" x="0" y="0" z="0"/>
                 </physvol>
        </loop>
        <auxiliary auxtype="Visibility" auxvalue="0"/>
  </volume>
```

To define in define block

Function of copyNo

....continue.....

....continue.....

```
<loop for="j" to="NumberOfReplicas" step="1">
     <volume name="BoxVolume[j]">
     <materialref ref="Material"/>
     <solidref ref="Box[j]"/>
     </volume>
</loop>

<volume name="World">
      <materialref ref="G4_Galactic"/>
      <solidref ref="WorldBox"/>
       <physvol>
          <file name="filename.gdml" />
          <positionref ref=”poscenter"/>
        </physvol>
        <physvol>
            <volumeref ref="TubeStack"/>
            <positionref ref= " poscenter" />
             <rotationref ref="norot"/>
        </physvol>
        <auxiliary auxtype="Visibility" auxvalue="0"/>
        <auxiliary auxtype="Region" auxvalue="WorldRegion,0.1"/>
</volume>

</structure>
```

Detector Construction - GDML

How does G4 call the physical volume we have defined?

‹volume name="TubeVolume"› → **TubeVolume_PV**[0]

‹volume name="BoxVolume[j]"› → **BoxVolume_j_PV**[0]

```
‹loop for="copyNo" to= " 2" step="1"›
   ‹physvol copynumber="copyNo"›
   ‹volumeref ref="TubeVolume"/›
   …
   ‹/physvol›
‹/loop›
```

→ **TubeVolume_PV**[copyNo]

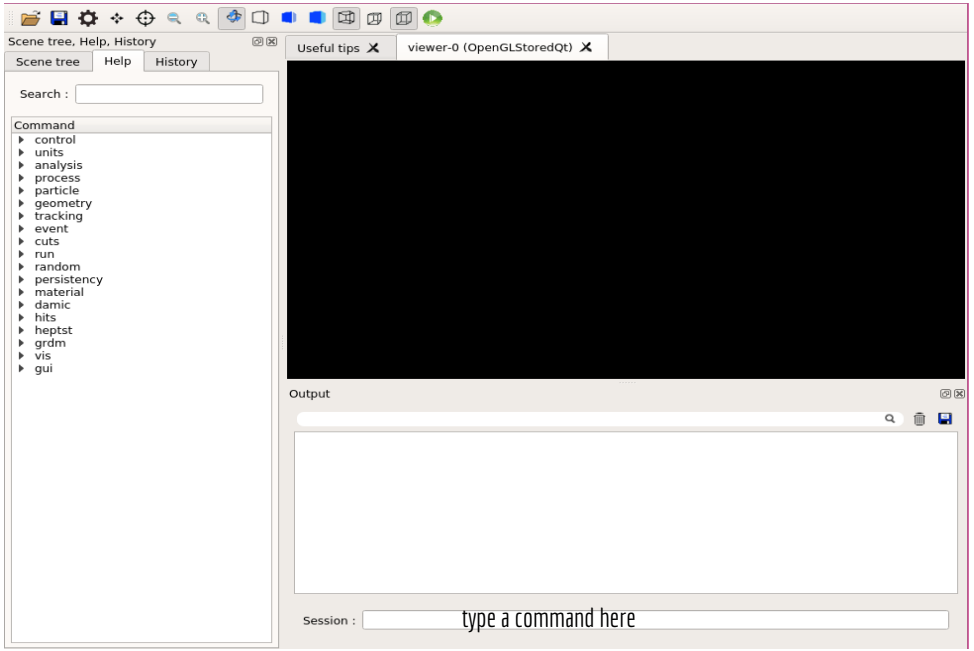
Sensitive detector

Given geometry, physics and primary track generation, Geant4 does proper physics simulation "silently". You have to add a bit of code to extract information useful to you.

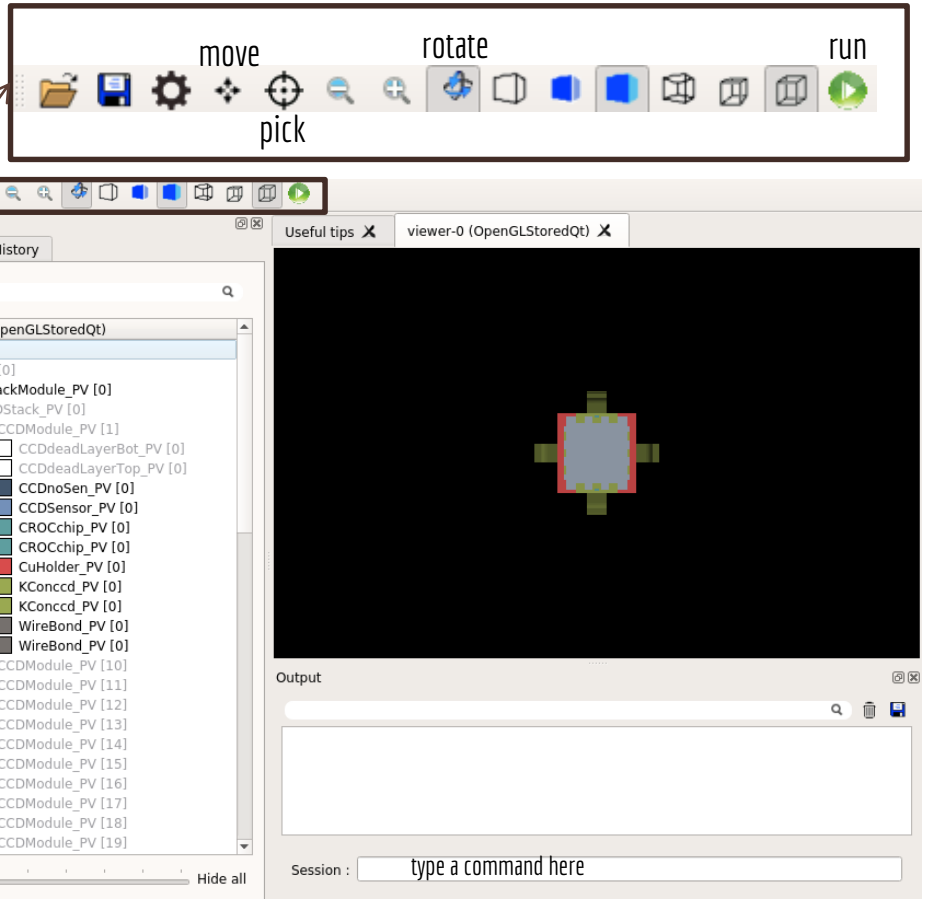
- A special user class, **sensitive detector:** attached to (a) selected logical volume(s), invoked when a step takes place in the sensitive logical volume
- A sensitive detector either constructs one or more hit objects or accumulates values to existing hits using information given in a G4Step object
- **Hit**: snapshot of the physical interaction of a track or an accumulation of interactions of tracks in the sensitive region
- Hit objects must be stored in a collection which will be associated to a G4Event
- DAMICM sensitive detector: CCD Sensor (look file DAMICCCDSD.cc)

Geometry check

- Open Interactive window In build directory: ./DAMICG4 file.gdml
- Geometry Visualization
- Check overlaps btw volumes:
  - Important to check: could cause Geant4 to behave in unpredictable ways!!!!!!
  - /geometry/test/run
- Check on terminal the physical Volumes when you launch simulations



Geometry visualization



Open graphics system:

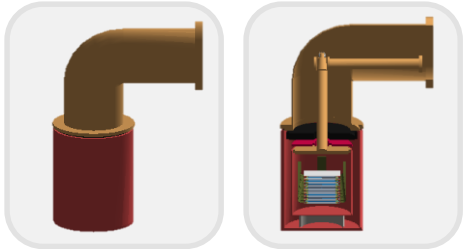
/vis/open OGL

now you can draw the detector:

/vis/drawVolume

Additional useful command:

/vis/viewer/addCutawayPlane 0 0 0 mm 0 1 0

before  after

Geometry check on terminal

When you launch simulation, on terminal:

--- List of PV:
Tag PV name N_instances Density Mass Volume Surface
[units] - - - g/cm^3 kg cm^3 cm^2
9 , AddEFCuShield_PV , 1 , 8.96 , 2.65086 , 295.855 , 674.22
8 , AddPbShield_PV , 1 , 11.35 , 6.71288 , 591.443 , 755.775
14 , AncientLeadTop_PV , 1 , 11.35 , 24.1737 , 2129.84 , 1866.63
19 , CCDSensor_PV , 50 , 2.33 , 0.636964 , 273.375 , 8221.5
20 , CCDnoSen_PV , 50 , 2.33 , 0.195002 , 83.6921 , 7791.82
27 , CROCchip_PV , 100 , 2.97 , 0.00169973 , 0.5723 , 50.674
6 , ColdShield_PV , 1 , 8.96 , 4.75604 , 530.808 , 3537.37
2 , Cryo_PV , 1 , 8.96 , 40.4032 , 4509.29 , 11504.3
15 , CuHolder_PV , 50 , 8.96 , 3.58681 , 400.313 , 25982
53 , HorizontalFinger_PV , 1 , 8.96 , 32.1862 , 3592.21 , 2123.86
28 , KCext_0_PV , 4 , 6.698 , 0.0146023 , 2.1801 , 135.574
38 , KCext_10_PV , 4 , 6.698 , 0.0260918 , 3.89546 , 239.431
39 , KCext_11_PV , 4 , 6.698 , 0.0271184 , 4.04874 , 253.491
40 , KCext_12_PV , 4 , 6.698 , 0.0285898 , 4.26841 , 264.325
41 , KCext_13_PV , 4 , 6.698 , 0.0295559 , 4.41265 , 272.267

Output

Output file: file_name.root (command: /analysis/setFileName file_name)

Output optimized for disk consumption and redundancy. Same variables used to link trees (eventID, trackID).

Trees in G4 simulation:

- **RunInfo**: general information on the simulation setup.
- **EventOut**: information on the generated event.
- **CCDOut**: Contains information on the energy deposits in the CCDs sensitive region.

Optional (you can switch on through command)

- **TrackOut:** Contains information on each step of the particle. (command: /damic/analysis/setstoretrack true)
- **PartInfo:** Contains information for each particle of a given event. The tree is filled once per event, with vectors of particle information. (command: /damic/analysis/setstoreparticleinfo true)

Information about variables inside trees: https://gitlab.in2p3.fr/damicm/DAMICM_G4Sims/-/wikis/G4%20Output

RunInfo tree

General information on the simulation setup.

branch	type	unit	description
NEvt	`int`	-	number of events simulated
NCCD	`int`	-	number of CCD in the detector geometry
Seed	`int`	-	Initial seed: -1 if base on the process time
CCDVersion	`int`	-	configuration ID of the CCD (see DB)
CCDFrameVersion	`int`	-	configuration ID of the Frame, 0 if OFF
CableVersion	`int`	-	configuration ID of the Cable, 0 if OFF
VesselVersion	`int`	-	configuration ID of the Vessel, 0 if OFF
CryoVersion	`int`	-	configuration ID of the Cryostat, 0 if OFF
ShieldingVersion	`int`	-	configuration ID of the Shielding, 0 if OFF
concatedVolumeNames	`string`	-	list of volumes in geometry
volumeNameID	`int`	-	Volume ID in the list
volumeMass	`double`	`kg`	mass of the volumes in the list
volumeDensity	`double`	`g/cm^3`	density of the volumes in the list
volumeVolume	`double`	`cm^3`	volume of volumes in the list
volumeSurface	`double`	`cm^2`	surface of the volumes in the list
primaryParticle	`string`	-	selected primary particles
primaryIon	`string`	-	primary ion A and Z (AaZz)
simulatedVolume	`string`	-	Logical volume in which the primary particles were simulated

EventOut tree

Information on the generated event.
It is filled once per event and contains NEvt entries in total.

branch	type	unit	description
EventID	int	-	EventID
pdg	int	-	pdg code of the primary particle
charge	int	-	charge of the primary particle
volid	int	-	volume ID of the generated particle
energy	double	eV	kinetic energy of the primary
momx	double	eV	x-coord of the primary particle's momentum
momy	double	eV	y-coord of the primary particle's momentum
momz	double	eV	z-coord of the primary particle's momentum
posx	double	mm	absolute position x (in G4 coordinate system)
posy	double	mm	absolute position y (in G4 coordinate system)
posz	double	mm	absolute position z (in G4 coordinate system)
triggerTime	double	s	trigger time

CCDOut tree

Contains information on the energy deposits in the CCDs sensitive region.
Tree is filled once per particle per pixel.
Read from Pyhton code.

branch	type	unit	description
EventID	int	-	EventID
pdg	int	-	pdg code of the particle
trackid	int	-	trackID of the particle
parentid	int	-	trackID of the mother particle (=0 if the particle is a primary)
CCDid	int	-	CCD id
posx	double	mm	edep-weighted average x-position per pixel (local pos)
posy	double	mm	edep-weighted average y-position per pixel (local pos)
posz	double	mm	edep-weighted average z-position per pixel (local pos)
gposx	double	mm	edep-weighted average x-position per pixel (global pos)
gposy	double	mm	edep-weighted average y-position per pixel (global pos)
gposz	double	mm	edep-weighted average z-position per pixel (global pos)
Edep	double	eV	total deposited energy in the pixel
time	double	s	global time
localtime	double	s	local time (wrt track time)

Useful Links

- GEANT4:

https://geant4.web.cern.ch/support

examples: https://geant4-userdoc.web.cern.ch/Doxygen/examples_doc/html/index.html

- GDML:

https://gdml.web.cern.ch/GDML/doc/GDMLmanual.pdf

- DAMICM G4 CODE:

https://gitlab.in2p3.fr/damicm/DAMICM_G4Sims

wiki: https://gitlab.in2p3.fr/damicm/DAMICM_G4Sims/-/wikis/home

If you want to partecipate to the analysis meetings, write an email to mariangela.settimo@subatech.in2p3.fr

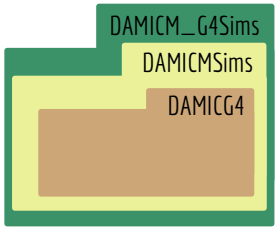
Questions???

Exercises

List of the exercises

- **Ex0:** download and compile the code
- **Ex1**: Launch simple simulations with already implemented gdml in batch mode and interactive mode (optional), inspect output trees
- **Ex2**: Build your own GDML, launch simulations, inspect output trees
- All the files needed for the exercises are in the directory **/sps/hep/damic/school/G4_exercises** on the Lyon cluster

Software Download



- You can download the code from git lab:

  git clone https://gitlab.in2p3.fr/damicm/DAMICM_G4Sims.git

- Different branches:

  git checkout namebranch

  namebranch: - for DAMICM Simulation: **v1.1.0**

DAMICM G4 code Compilation

- **Source root and geant4**,

  example on Lyon Cluster: ccenv root 6.18.00

  ccenv geant4 10.04.p02

  } Each time you login to CCIN2P3

- **Compilation:**
  - **Create directory build in DAMICMSims**: mkdir build
  - **Go in build**: cd build
  - **Compile the code**: cmake ../DamicG4/ -DCMAKE_INSTALL_PREFIX=$PWD

    make install
  - Each time you modify the code you have to recompile in the directory «build» typing «make install»

First exercise

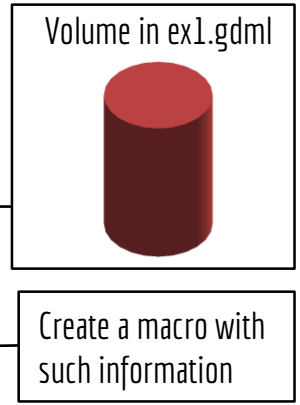
Launch a simulation (in build directory)

1. In **batch mode**:

   - use ex1.gdml
     (put this in a gdml directory in build: ./DAMICG4 gdml/ex1.gdml file.mac)

   - run 1 event with: e- of 20 MeV, direction (0,0,-1), point source at (0,0, 500 mm)

   - look at output file test.root (default output file name, if no name specified)



First exercise

2. OPTIONAL:

In **interactive mode**:

- use gdml: ex1.gdml

- execute macro ex1_vis.mac to set visualization options* (/control/execute ex1_vis.mac)

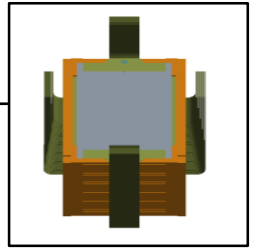
- run 1 event with the same information given in point 1

(*) for more visualization options look at vis.mac in DAMICG4 directory

First exercise

3. Run again the same simulation in batch mode but with another gdml (with DAMICM CCDStack):

- use detector_main.gdml* file in DamicG4/gdml directory
  copy gdml directory in build directory!



- run 1 event with: e- of 20 MeV, direction (0,0,-1), point source at (0,0, 500 mm)

Create a macro with such information

- look at output file test.root (change the output name if you don't want to overwrite the previous file)

- What did change with respect to the other simulation?

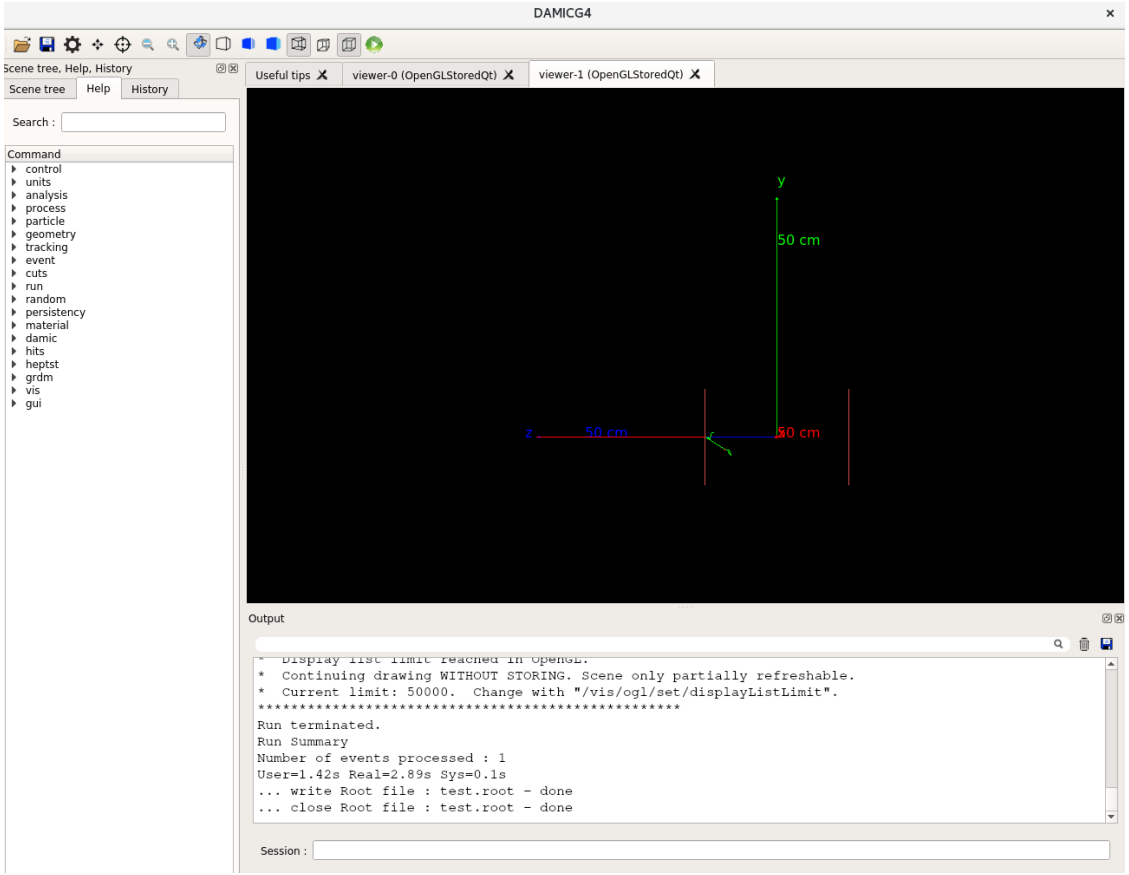
(*) the detector_main calls another gdml file that is: CCDModule_v20190211_placement.gdml!

First exercise

trajectories are color-coded by charge:

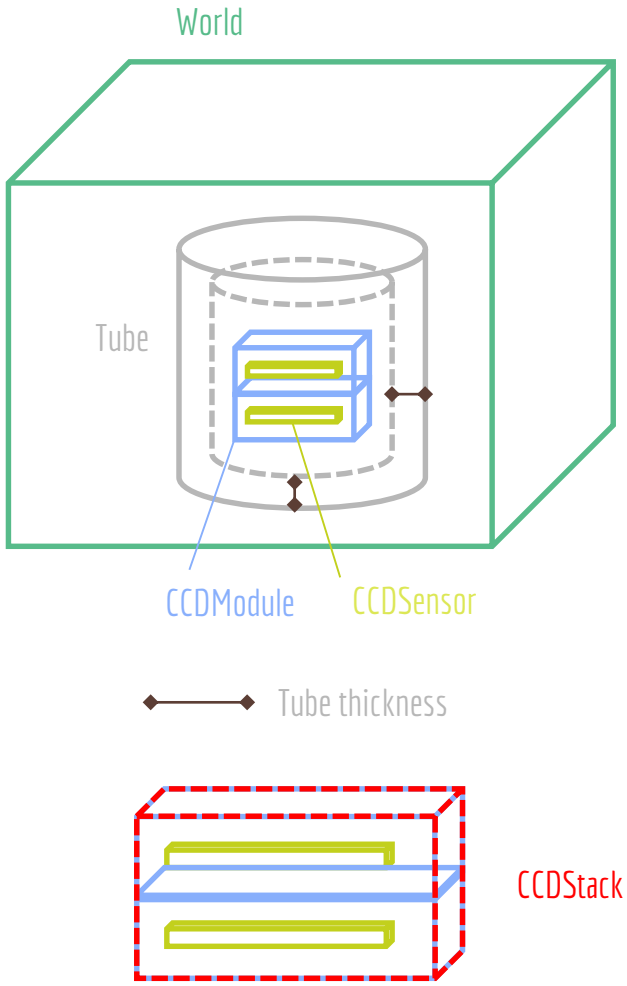
- positive = blue
- neutral = green
- negative = red

Output file opened by default:
test.root



Second exercise

- Write a simple gdml file (ex2_tocomplete.gdml) with following volumes:
  - World: Box 5m x 5m x 5m (G4_Galactic)
  - Tube: Tube Rmax: 15 cm, Heigh:30 cm, thickness: 0.5 cm (Copper)
  - CCDSensor (Sensitive Volume): 9 cm x 9 cm x 0.675 mm (Silicon)
  - CCDModule: 10 cm x 10 cm x 1 mm (G4_Galactic) with inside the CCDSensor (in the center). Create 2 identical CCDModule, assign a copy number to each of them, put them on top of each other (you have to create a Stack that contains the 2 modules)

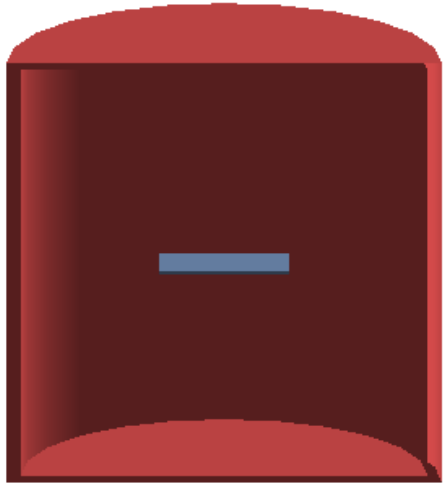


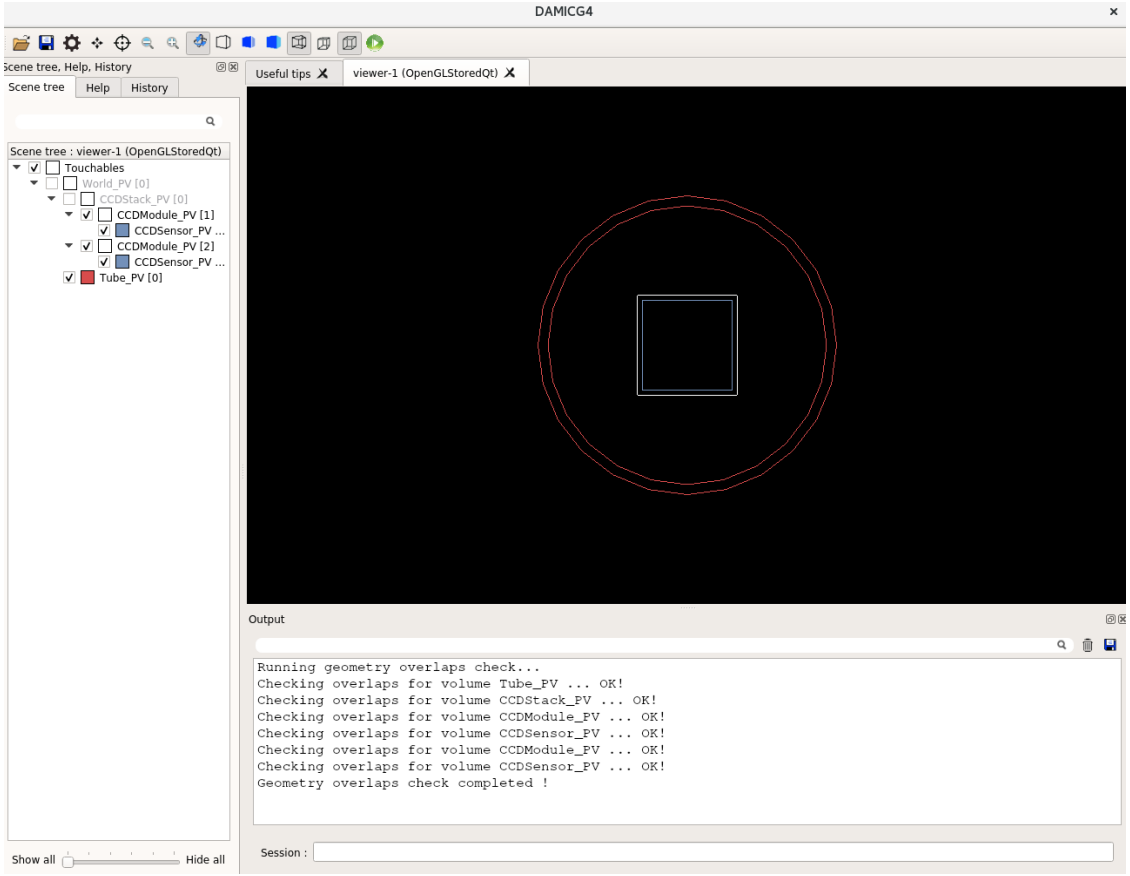
Second exercise

- Check the geometry and the overlaps with the interactive window
- Launch a simulation (**BATCH MODE**, look file.mac slide 9) with:
    - primary particle: ion A:208 Z:81 ,
    - particle source: Tube Volume
    - Nevts: 1000
    - activate optional trees
- Look at the output trees
- Check in the EventOut the position of the primary particles: plot of x,y,z position
- Do the plot of the Edep in the CCDOut

- Run again the simulation this time replacing the CCDStack with the file CCDModule_v20190211_placement.gdml, (Delete or comment the lines of code of the previous CCDStack,CCDModule,Sensor)

Second exercise





Backup

Physics list

Useful link for Physics List:

- https://geant4.web.cern.ch/node/1619 (Livermore)
- https://geant4.web.cern.ch/node/1621 (Penelope)