Hands-on:
Dosimetry in a water phantom

Geant4-DNA

A SIMULATION TOOLKIT

geant4-dna.org

Dousatsu SAKATA
AIST, Tsukuba, Japan
dousatsu.sakata@aist.go.jp

Geant4-DNA tutorial
Thaïlande
18-20/3/2026

**Geant4 version 11.4
Released in December 2025**

How to download the application 1/2

Acess to the tutorial page from the browser on your VMware

How to download the application 2/2

③

Enter the password
2026BangkokTutorial#

④

⑤

The file will be saved in ~/Downloads (/home/local1/Downloads)

medical_linac
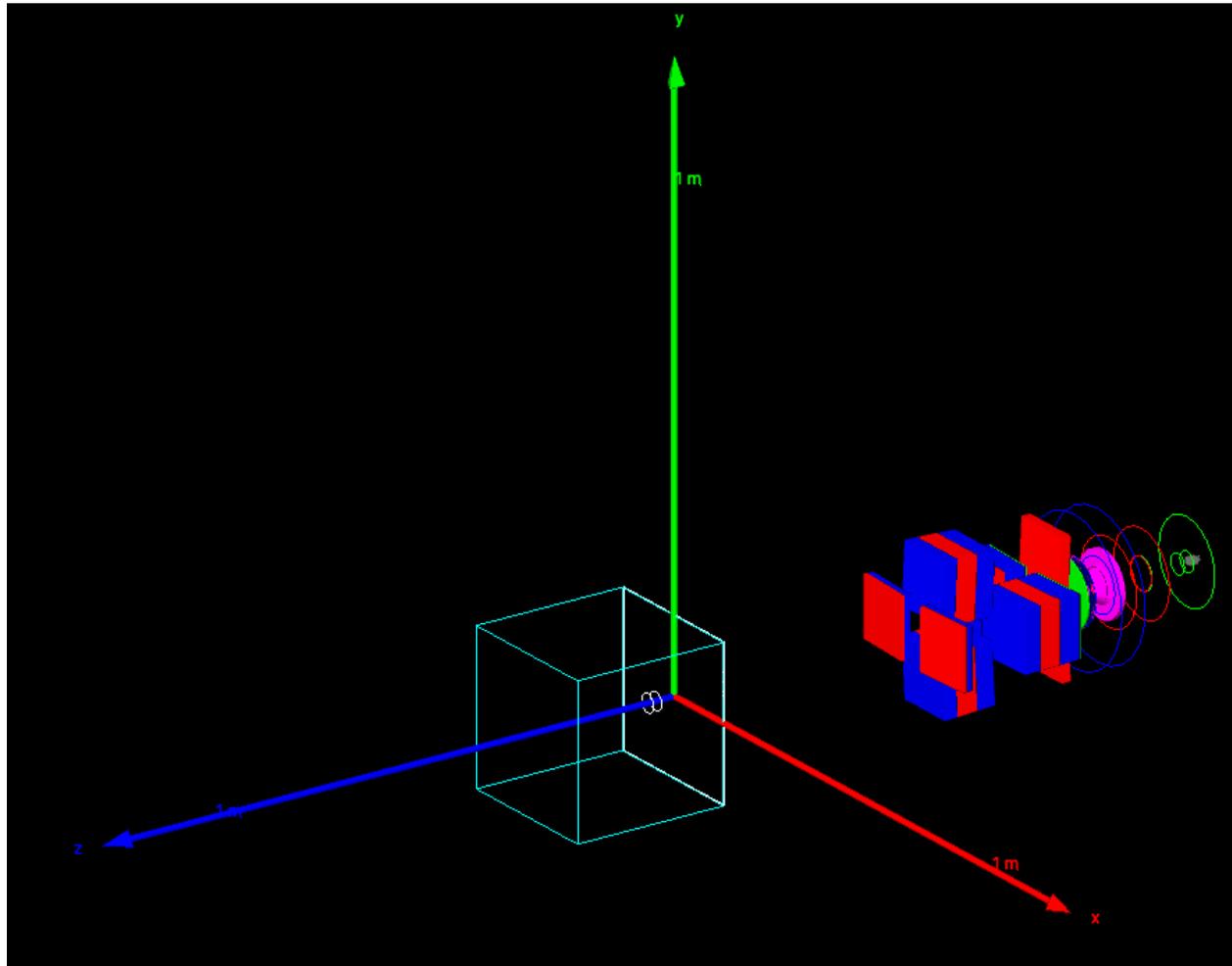
This application based on examples/advanced/medical_linac.

The geometry imitates GE Saturn 43 LINAC

Excises in this hands-on

1: How to compile and run Geat4 application

2: How to handle IAEA phase space file

3: How to score dose depth curve
   (Using Built-in scoring interface)

4: Advanced ! : we don't support in this tutorial
   How to score absorbed dose in an ionization chamber
   (Using SensitiveDetector via MultiFunctionalDetector)

Step1 – 1 : Build Application

① Execute the following commands from terminal

```
$ mkdir G4Work
$ cd G4Work
$ cp –r  ~/Downloads/medical_linac_handson.tar.gz   .
$ tar -zxvf medical_linac_handson.tar.gz
$ cd medical_linac_handson
$ mkdir build
$ cd build
$ cmake ../
$ make
$ ./medical_linac
```
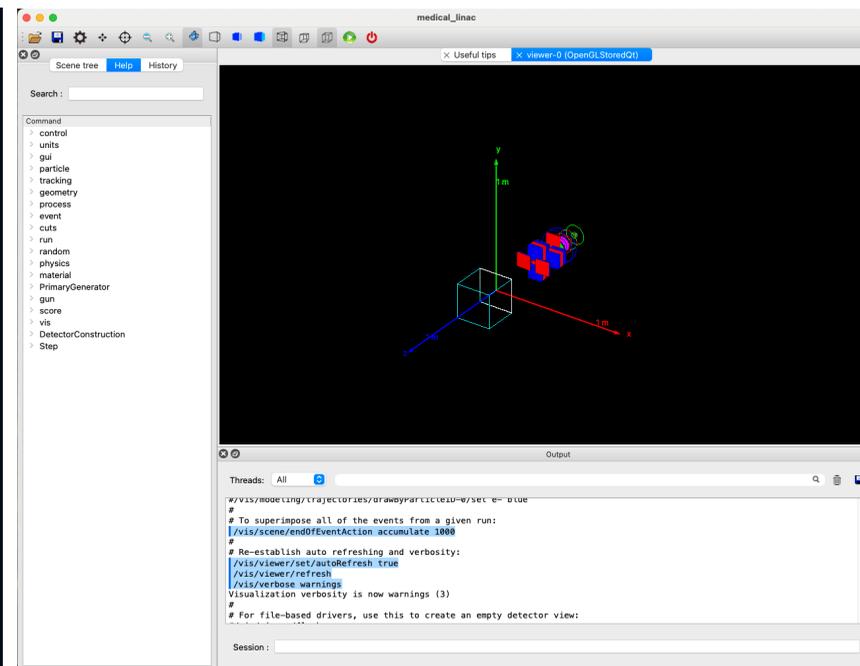

UI window will be popped up

Step1 – 2 : Execute command from UI

② Execute the built-in command from the UI window
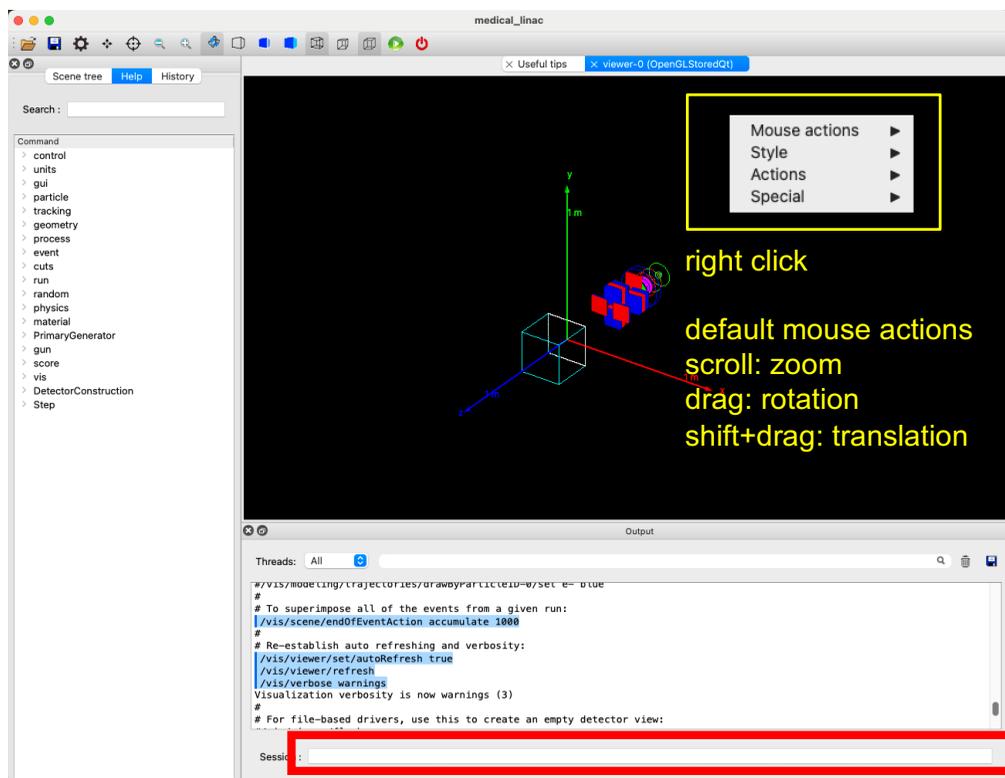
/run/beamOn 100 and enter
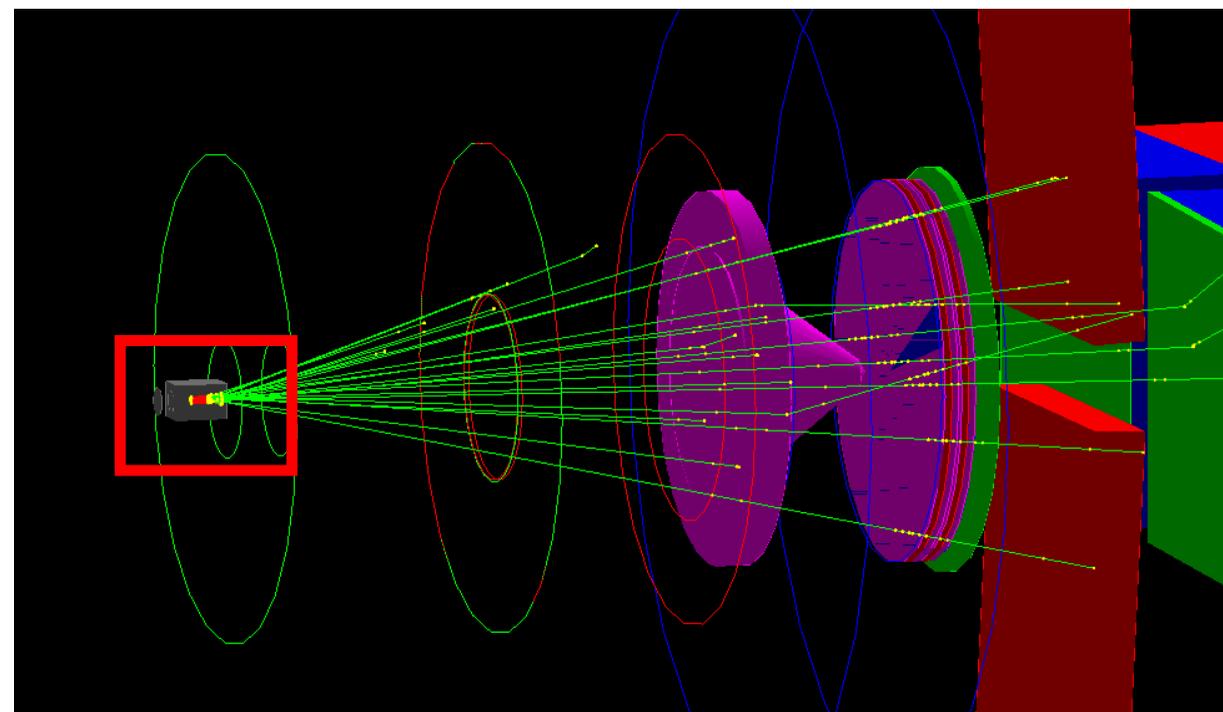
Irradiate 100 electrons to target
Red: electron
Green: gamma

How to edit text files?

■ gedit

$ gedit run.mac

Save Close

■ nano

$ nano run.mac

Save: Ctrl+o → Enter
Close: Ctrl+x

GEANT4-DNA

Step1 – 3 : Execute with macro

③ Control number of events from macro file
change the following line in run.mac

```
#### verbosities
/control/verbose 1
/material/verbose 0
/run/verbose 0
/process/em/verbose 0

#/run/numberOfThreads 4

/physics/addPhysics standard_opt3

#/PrimaryGenerator/useiaeaFile true                          # handson S2
#/PrimaryGenerator/iaeaFileName IAEAPSF/VarianClinaciX_6MV_10x10_w1 # handson S2
#/PrimaryGenerator/gunOffset -23.2 cm                        #  -(90.-66.8)*cm # handson S2

/run/initialize

#/DetectorConstruction/PlateIonChamber/sensitiveMaterial G4_AIR  # handson S4
#/DetectorConstruction/PlateIonChamber/sensitiveMaterial G4_WATER # handson S4

#/score/create/boxMesh boxMesh_1                     # handson S3
#/score/mesh/translate/xyz 0 0 130 mm               # handson S3
#/score/mesh/boxSize 127.5 127.5 127.5 mm  # half dim  # handson S3
#/score/mesh/nBin 51 51 51                           # handson S3
#/score/quantity/doseDeposit dDep                    # handson S3
#/score/close                                        # handson S3
#/score/list                                         # handson S3

#
#/run/printProgress 100000
/run/beamOn 1000  # handson S1
#/run/beamOn 33333300 # handson S2 for IAEAPSF/VarianClinaciX_6MV_10x10_w1

#/score/drawProjection boxMesh_1 dDep
#/score/dumpQuantityToFile boxMesh_1 dDep dDep.txt # handson S3
```
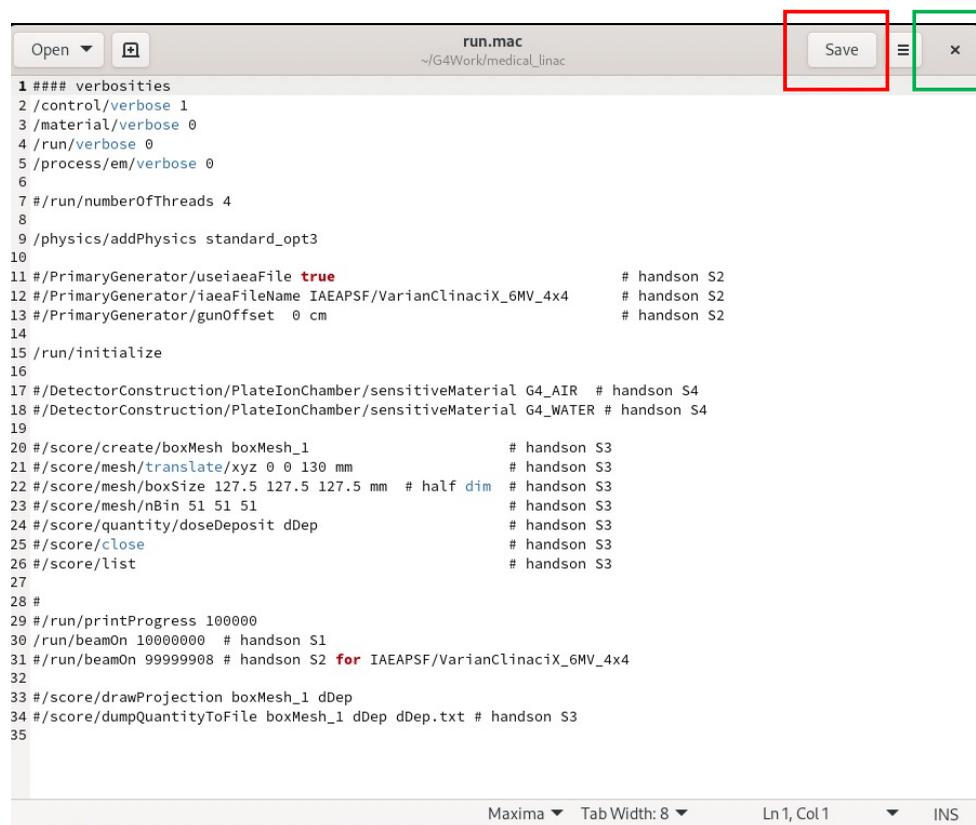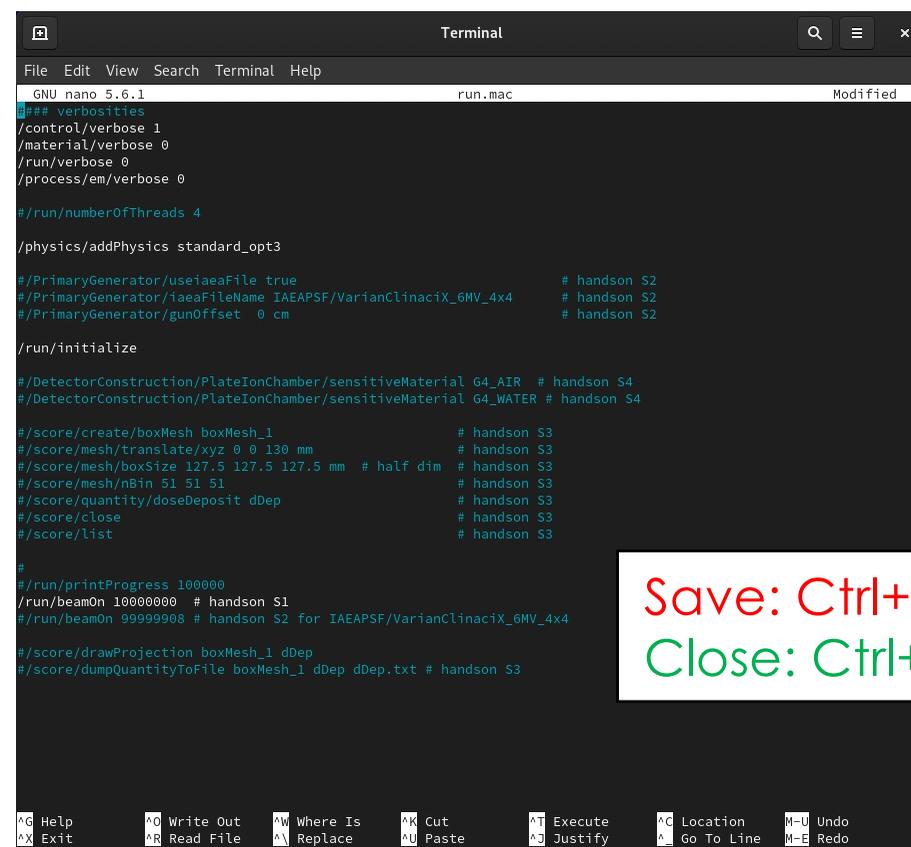
④ Execute the command from terminal

$ ./medical_linac run.mac

```
hPairProd:  for pi+  XStype:1  SubType=4
        dE/dx and range tables from 10 eV  to 100 TeV in 260 bins
        Lambda tables from threshold to 100 TeV, 20 bins/decade, spline: 1
        Sampling table 20x1001; from 1.11656 GeV to 100 TeV
        ===== EM models for the G4Region  DefaultRegionForTheWorld ======
            hPairProd : Emin=     0 eV  Emax=  100 TeV  ModifiedMephi

msc:  for pi-  SubType= 10
        ===== EM models for the G4Region  DefaultRegionForTheWorld ======
            UrbanMsc : Emin=     0 eV  Emax=  100 TeV Nbins=240 100 eV  - 100 TeV
        StepLim=Minimal Rfact=0.2 Gfact=2.5 Sfact=0.6 DispFlag:1 Skin=1 Llim=1 mm

hIoni:  for pi-  XStype:3  SubType=2
        dE/dx and range tables from 10 eV  to 100 TeV in 260 bins
        Lambda tables from threshold to 100 TeV, 20 bins/decade, spline: 1
        StepFunction=(0.2, 0.05 mm), integ: 3, fluct: 1, linLossLim= 0.01
        ===== EM models for the G4Region  DefaultRegionForTheWorld ======
            ICRU73QO : Emin=     0 eV  Emax=297.505 keV  deltaVI
            BetheBloch : Emin=297.505 keV Emax=  100 TeV  deltaVI

hBrems:  for pi-  XStype:1  SubType=3
        dE/dx and range tables from 10 eV  to 100 TeV in 260 bins
        Lambda tables from threshold to 100 TeV, 20 bins/decade, spline: 1
        ===== EM models for the G4Region  DefaultRegionForTheWorld ======
            hBrem : Emin=     0 eV  Emax=  100 TeV  ModifiedMephi

hPairProd:  for pi-  XStype:1  SubType=4
        dE/dx and range tables from 10 eV  to 100 TeV in 260 bins
        Lambda tables from threshold to 100 TeV, 20 bins/decade, spline: 1
        Sampling table 20x1001; from 1.11656 GeV to 100 TeV
        ===== EM models for the G4Region  DefaultRegionForTheWorld ======
            hPairProd : Emin=     0 eV  Emax=  100 TeV  ModifiedMephi

                        End of Global Run----------------------
    The run was 1000 events
Graphics systems deleted.
Visualization Manager deleting...
```

See the number of events

Step2 – 0 : IAEA phase space file (PSF)

IAEA provides phase space file for several medical linacs

https://www-nds.iaea.org/phsp/phsp.htmlx

In this application these files stored under IAEAPSF
VarianClinaciX_6MV_4x4.IAEAheader(Text file for PSF detail)
VarianClinaciX_6MV_4x4.IAEAphsp (Binary PSF file)

Step2 – 1 : How to handle IAEA phase space file (PSF)

Source files to read IAES phase space file

in medical_linac_handson/include
G4IAEAphspReader.hh
iaea_config.h
iaea_header.h
iaea_phsp.h
iaea_record.h
utilities.h

in medical_linac_handson/src
G4IAEAphspReader.cc
iaea_config.cc
iaea_header.cc
iaea_phsp.cc
iaea_record.cc
utilities.cc

Geant4 class for PSF handler

IAEA PSF handler

In step2, you will be trying to implement the application as an application developer.

```
dousatsu@dousatsunoMacBook-Pro include % grep S2 *    You can check the lines involved in step 2
G4IAEAphspReader.hh:  //{ G4Exception("Cannot use G4IAEAphspReader void constructor"); } // handson S2 comment out
G4IAEAphspReader.hh:           { G4Exception("Cannot use G4IAEAphspReader void constructor","", JustWarning,""); } //handson S2
G4IAEAphspReader.hh:    //if (parallelRun > theTotalParallelRuns && parallelRun < 1) G4Exception("Error in G4IAEAphspReader::SetParallelRun()"); //handson S2 comment out
G4IAEAphspReader.hh:    if (parallelRun > theTotalParallelRuns && parallelRun < 1)  { G4Exception("Cannot use G4IAEAphspReader void constructor","", JustWarning,""); } //handson S2
G4IAEAphspReader.hh:           //inline void SetFileName(G4String name)  {theFileName=name;}                                          // handson S2
G4IAEAphspReader.hh:  //inline void SetGunOffset(G4double val)  {gunoffset=val;theGlobalPhspTranslation ={0,0,gunoffset};} // handson S2
G4IAEAphspReader.hh:           G4double gunoffset;// handson S2
PrimaryGeneratorAction.hh://class G4IAEAphspReader; // handson S2
PrimaryGeneratorAction.hh: //void UseIAEAFile    (G4bool value);   // handson S2
PrimaryGeneratorAction.hh: //void SetIAEAFileName(G4String value); // handson S2
PrimaryGeneratorAction.hh:      //void SetGunOffset(G4double value);    // handson S2
PrimaryGeneratorAction.hh:      //G4IAEAphspReader* theIAEAReader; // handson S2
PrimaryGeneratorAction.hh: //G4bool useiaeaFile;              // handson S2
PrimaryGeneratorAction.hh: //G4String theFileName;            // handson S2
PrimaryGeneratorAction.hh:      //G4double gunOffset;              // handson S2
PrimaryGeneratorMessenger.hh://#include <G4UIcmdWithABool.hh> // handson S2
PrimaryGeneratorMessenger.hh:  //G4UIcmdWithABool*         cmduseIAEA;  // handson S2
PrimaryGeneratorMessenger.hh:  //G4UIcmdWithAString*        cmdFileName; // handson S2
PrimaryGeneratorMessenger.hh:          //G4UIcmdWithADoubleAndUnit* cmdGunOffset;// handson S2
dousatsu@dousatsunoMacBook-Pro include %
```

⑤ Try to call G4IAEAphspReader in PrimaryGeneratorAction !

PrimaryGeneratorAction.hh

```cpp
#ifndef PRIMARY_GENERATOR_ACTION_HH
#define PRIMARY_GENERATOR_ACTION_HH

#include <G4VUserPrimaryGeneratorAction.hh>
#include <G4ParticleDefinition.hh>

#include "PrimaryGeneratorMessenger.hh"
#include "RunAction.hh"

class G4ParticleGun;
class PrimaryGeneratorMessenger;
class G4IAEAphspReader; // handson S2

class PrimaryGeneratorAction : public G4VUserPrimaryGeneratorAction
{
public:
 PrimaryGeneratorAction();
 ~PrimaryGeneratorAction();
 void GeneratePrimaries(G4Event* anEvent) override;
 void GenerateFromRandom();

 void SetGunRadius(G4double value);
 void SetGunMeanEnergy(G4double value);
 void SetGunStdEnergy(G4double value);
 void SetSource(G4String value);
 //void UseIAEAFile    (G4bool value);   // handson S2
 //void SetIAEAFileName(G4String value); // handson S2
 //void SetGunOffset(G4double value);    // handson S2
```

keep these comment out the three lines

```cpp
 private:

 G4ParticleGun* fGun;

 G4double gunRadius, gunMeanEnergy, gunStdEnergy;
 G4double energy;

 G4ParticleDefinition* particle;
 PrimaryGeneratorMessenger* priGenMessenger;
 G4ThreeVector position, direction;

 G4IAEAphspReader* theIAEAReader; // handson S2
 G4bool useiaeaFile;              // handson S2
 G4String theFileName;            // handson S2
 G4double gunOffset;              // handson S2
};

#endif
```

PrimaryGeneratorAction.cc

```cpp
#include <G4GeneralParticleSource.hh>
#include <Randomize.hh>
#include <G4RunManager.hh>
#include <G4PhysicalConstants.hh>

#include "G4IAEAphspReader.hh" // handson S2

using namespace std;

PrimaryGeneratorAction::PrimaryGeneratorAction()
: fGun(nullptr),
  useiaeaFile(false),theFileName(""),theIAEAReader(nullptr), gunOffset(0),//handson S2
  gunRadius(0), gunMeanEnergy(0), gunStdEnergy(0),
  energy(0),
  particle(nullptr)

{
  priGenMessenger = new PrimaryGeneratorMessenger(this);
  G4int n_particle = 1;
  fGun = new G4ParticleGun(n_particle);
  theIAEAReader = new G4IAEAphspReader(theFileName); // handson S2
}

PrimaryGeneratorAction::~PrimaryGeneratorAction()
{
    delete priGenMessenger;
    delete fGun;

    if (theIAEAReader) { delete theIAEAReader; } // handson S2
}

void PrimaryGeneratorAction::GeneratePrimaries(G4Event* anEvent)
{
  if(useiaeaFile){                                     // handson S2
    theIAEAReader->SetFileName(theFileName);           // handson S2
    theIAEAReader->SetGunOffset(gunOffset);            // handson S2
    theIAEAReader->GeneratePrimaryVertex(anEvent);     // handson S2
  }else{                                               // handson S2
    GenerateFromRandom();
    fGun -> SetParticlePosition(position);
    fGun -> SetParticleMomentumDirection((G4ParticleMomentum)direction);
    fGun -> SetParticleEnergy(energy * MeV);
    fGun -> SetParticleDefinition(particle);
    // create vertex with previous specifications
    fGun -> GeneratePrimaryVertex(anEvent);
  }                                                    // handson S2
}
```

# Example of command to edit a file
$ gedit ../src/PrimaryGeneratorAction.cc

⑥ Make again and execute the command

$ make
$ ./medical_linac run.mac

Still running without PSF...
Why?
→ useiaeaFile = false

Let's try to control the flag by macro command!

Step2 – 3 : Setup Messenger Class ①

⑦ Try to implement class member functions

PrimaryGeneratorAction.hh   PrimaryGeneratorAction.cc

```cpp
#ifndef PRIMARY_GENERATOR_ACTION_HH
#define PRIMARY_GENERATOR_ACTION_HH

#include <G4VUserPrimaryGeneratorAction.hh>
#include <G4ParticleDefinition.hh>

#include "PrimaryGeneratorMessenger.hh"
#include "RunAction.hh"

class G4ParticleGun;
class PrimaryGeneratorMessenger;
class G4IAEAphspReader; // handson S2

class PrimaryGeneratorAction : public G4VUserPrimaryGeneratorAction
{
public:
 PrimaryGeneratorAction();
 ~PrimaryGeneratorAction();
 void GeneratePrimaries(G4Event* anEvent) override;
 void GenerateFromRandom();

 void SetGunRadius(G4double value);
 void SetGunMeanEnergy(G4double value);
 void SetGunStdEnergy(G4double value);
 void SetSource(G4String value);
 void UseIAEAFile    (G4bool value);   // handson S2
 void SetIAEAFileName(G4String value); // handson S2
 void SetGunOffset(G4double value);    // handson S2


private:

 G4ParticleGun* fGun;

 G4double gunRadius, gunMeanEnergy, gunStdEnergy;
 G4double energy;

 G4ParticleDefinition* particle;
 PrimaryGeneratorMessenger* priGenMessenger;
 G4ThreeVector position, direction;


 G4IAEAphspReader* theIAEAReader; // handson S2
 G4bool useiaeaFile;             // handson S2
 G4String theFileName;           // handson S2
 G4double gunOffset;             // handson S2
};

#endif
```

~/Work/AIST/G4Work/medical_linac_handson/include/PrimaryGeneratorAction.hh

```cpp
void PrimaryGeneratorAction::GenerateFromRandom()
{
  DetectorConstruction* detector = (DetectorConstruction*)
    G4RunManager::GetRunManager() -> GetUserDetectorConstruction();

  G4double zPos = -(detector -> GetAccOriginPosition()) - 5.*mm; // 5mm before the target disk
  G4double alpha, rho, phi, sinTheta, cosTheta;

  sinTheta = G4RandGauss::shoot(0., 0.003);
  cosTheta = sqrt(1-sinTheta*sinTheta);
  phi = twopi * G4UniformRand();
  rho = gunRadius * G4UniformRand();
  alpha = twopi * G4UniformRand();

  particle = G4ParticleTable::GetParticleTable()->FindParticle("e-");
  direction = {sinTheta * cos(phi), sinTheta * sin(phi), cosTheta};
  position = {rho * sin(alpha), rho * cos(alpha), zPos};
  energy = G4RandGauss::shoot(gunMeanEnergy, gunStdEnergy);
}

void PrimaryGeneratorAction::SetGunRadius(G4double value)
{
  gunRadius = value;
}

void PrimaryGeneratorAction::SetGunMeanEnergy(G4double value)
{
  gunMeanEnergy = value;
}

void PrimaryGeneratorAction::SetGunStdEnergy(G4double value)
{
  gunStdEnergy = value;
}
void PrimaryGeneratorAction::UseIAEAFile(G4bool value)       // handson S2
{                                                            // handson S2
  useiaeaFile = value;                                       // handson S2
}                                                            // handson S2
void PrimaryGeneratorAction::SetIAEAFileName(G4String value) // handson S2
{                                                            // handson S2
  theFileName = value;                                       // handson S2
}                                                            // handson S2
void PrimaryGeneratorAction::SetGunOffset(G4double value)    // handson S2
{                                                            // handson S2
  gunOffset = value;                                         // handson S2
}                                                            // handson S2
```

~/Work/AIST/G4Work/medical_linac_handson/src/PrimaryGeneratorAction.cc[+]

⑧ Check the application can be compiled without error

```
$ make
```

⑨ Implement use macro commands in PrimaryGeneratorMessenger

PrimaryGeneratorMessenger.hh   PrimaryGeneratorMessenger.cc

```cpp
#ifndef PRIMARY_GENERATOR_MESSENGER_HH
#define PRIMARY_GENERATOR_MESSENGER_HH

#include <G4UImessenger.hh>
#include <G4UIdirectory.hh>
#include <G4UIcmdWithADoubleAndUnit.hh>
#include <G4UIcmdWithABool.hh> // handson S2
#include <G4UIcmdWithAString.hh>
#include <G4SystemOfUnits.hh>

#include "PrimaryGeneratorAction.hh"
//#include "HistoManager.hh"

class PrimaryGeneratorAction;
class HistoManager;
class G4UIdirectory;
class G4UIcmdWithADoubleAndUnit;

class PrimaryGeneratorMessenger : public G4UImessenger
{
 public:
  PrimaryGeneratorMessenger(PrimaryGeneratorAction* );
  ~PrimaryGeneratorMessenger();
  void SetNewValue(G4UIcommand*, G4String) override;

 private:

  PrimaryGeneratorAction* primaryGenPointer;

  G4UIcmdWithADoubleAndUnit* cmdGunRadius;
  G4UIcmdWithADoubleAndUnit* cmdGunMeanEnergy;
  G4UIcmdWithADoubleAndUnit* cmdGunStdEnergy;
  G4UIcmdWithABool*          cmduseIAEA;  // handson S2
  G4UIcmdWithAString*        cmdFileName; // handson S2
  G4UIcmdWithADoubleAndUnit* cmdGunOffset;// handson S2

};

#endif /* PRIMARY_GENERATOR_MESSENGER_HH */
~
~
~
~
~
~
~
```

```cpp
#include "PrimaryGeneratorMessenger.hh"

PrimaryGeneratorMessenger::~PrimaryGeneratorMessenger()
{
  delete cmdGunRadius;
  delete cmdGunMeanEnergy;
  delete cmdGunStdEnergy;
  delete cmduseIAEA;   // handson S2
  delete cmdFileName;  // handson S2
  delete cmdGunOffset; // handson S2
}

PrimaryGeneratorMessenger::PrimaryGeneratorMessenger(PrimaryGeneratorAction* myGun)
:
  cmduseIAEA(nullptr),cmdFileName(nullptr), cmdGunOffset(nullptr), // handson S2
  primaryGenPointer(myGun), cmdGunRadius(nullptr),
  cmdGunMeanEnergy(nullptr), cmdGunStdEnergy(nullptr)
{
  cmdGunRadius = new G4UIcmdWithADoubleAndUnit("/PrimaryGenerator/gunRadius", this);
  cmdGunRadius -> SetDefaultUnit("mm");
  cmdGunRadius -> SetDefaultValue(1.);
  primaryGenPointer -> SetGunRadius(1.*mm);

  cmdGunMeanEnergy = new G4UIcmdWithADoubleAndUnit("/PrimaryGenerator/gunMeanEnergy", this);
  cmdGunMeanEnergy -> SetDefaultUnit("MeV");
  cmdGunMeanEnergy -> SetDefaultValue(12.);
  primaryGenPointer -> SetGunMeanEnergy(12.*MeV);

  cmdGunStdEnergy = new G4UIcmdWithADoubleAndUnit("/PrimaryGenerator/gunStdEnergy", this);
  cmdGunStdEnergy -> SetDefaultUnit("MeV");
  cmdGunStdEnergy -> SetDefaultValue(0.127);
  primaryGenPointer -> SetGunStdEnergy(0.127*MeV);

  cmduseIAEA  = new G4UIcmdWithABool("/PrimaryGenerator/useiaeaFile", this); // handson S2
  cmduseIAEA  -> SetDefaultValue(false);                         // handson S2
  cmduseIAEA  -> SetGuidance("flag to use IAEA phase space file");   // handson S2
  primaryGenPointer -> UseIAEAFile(false);                     // handson S2

  cmdFileName  = new G4UIcmdWithAString("/PrimaryGenerator/iaeaFileName", this);// handson S2
  cmdFileName  -> SetDefaultValue("IAEAPSF/Varian_Clinac_600C_6MV_10x10");    // handson S2
  cmdFileName  -> SetGuidance("Set the IAEA file name ");          // handson S2
  primaryGenPointer -> SetIAEAFileName("IAEAPSF/Varian_Clinac_600C_6MV_10x10"); // handson S2

  cmdGunOffset = new G4UIcmdWithADoubleAndUnit("/PrimaryGenerator/gunOffset", this);// handson S2
  cmdGunOffset -> SetDefaultUnit("mm");                        // handson S2
  cmdGunOffset -> SetDefaultValue(0.);                        // handson S2
  primaryGenPointer -> SetGunOffset(0.*mm);                    // handson S2
}
```

```cpp
void PrimaryGeneratorMessenger::SetNewValue(G4UIcommand* cmd, G4String newValue)
{
  if ( cmd == cmdGunRadius )
  {
    cmdGunRadius -> GetNewUnitValue(newValue);
    primaryGenPointer -> SetGunRadius(cmdGunRadius -> GetNewDoubleValue(newValue));
  }
  else if ( cmd == cmdGunMeanEnergy )
  {
    cmdGunMeanEnergy -> GetNewUnitValue(newValue);
    primaryGenPointer -> SetGunMeanEnergy(cmdGunMeanEnergy -> GetNewDoubleValue(newValue));
  }
  else if ( cmd == cmdGunStdEnergy )
  {
    cmdGunStdEnergy -> GetNewUnitValue(newValue);
    primaryGenPointer -> SetGunStdEnergy(cmdGunStdEnergy -> GetNewDoubleValue(newValue));
  }
  else if ( cmd == cmduseIAEA )          // handson S2
  {                                // handson S2
    primaryGenPointer -> UseIAEAFile(newValue);// handson S2
  }                                // handson S2
  else if ( cmd == cmdFileName )         // handson S2
  {                                // handson S2
    primaryGenPointer -> SetIAEAFileName(newValue);// handson S2
  }                                // handson S2
  else if ( cmd == cmdGunOffset )                              // handson S2
  {                                                       // handson S2
    cmdGunOffset -> GetNewUnitValue(newValue);
    primaryGenPointer -> SetGunOffset(cmdGunOffset -> GetNewDoubleValue(newValue));// handson S2
  }                                                       // handson S2
  else
    G4cerr << "PrimaryGeneratorMessenger::SetNewValue: command not found" << G4endl;
}
```

⑩ Check the application can be compiled without error

$ make

Step2 – 5 : control IAES PSF via macro commands

⑪ control the class member via macro commands

in run.mac

```
#### verbosities
/control/verbose 1
/material/verbose 0
/run/verbose 0
/process/em/verbose 0

#/run/numberOfThreads 4

/physics/addPhysics standard_opt3

/PrimaryGenerator/useiaeaFile true                      # handson S2
/PrimaryGenerator/iaeaFileName IAEAPSF/VarianClinaciX_6MV_4x4    # handson S2
/PrimaryGenerator/gunOffset  0 cm                       # handson S2

/run/initialize

#/DetectorConstruction/PlateIonChamber/sensitiveMaterial G4_AIR  # handson S4
#/DetectorConstruction/PlateIonChamber/sensitiveMaterial G4_WATER # handson S4

#/score/create/boxMesh boxMesh_1                  # handson S3
#/score/mesh/translate/xyz 0 0 130 mm            # handson S3
#/score/mesh/boxSize 127.5 127.5 127.5 mm  # half dim  # handson S3
#/score/mesh/nBin 51 51 51                       # handson S3
#/score/quantity/doseDeposit dDep               # handson S3
#/score/close                                    # handson S3
#/score/list                                     # handson S3

#
#/run/printProgress 100000
/run/beamOn 10000000  # handson S1
#/run/beamOn 99999908 # handson S2 for IAEAPSF/VarianClinaciX_6MV_4x4

#/score/drawProjection boxMesh_1 dDep
#/score/dumpQuantityToFile boxMesh_1 dDep dDep.txt # handson S3
~
```

activate IAEA PSF

set PSF file name

set Z position (see IAEAheader)

irradiate particles at z =0 cm

⑫ Execute the command

```
$ ./medical_linac run.mac
```

```
--------------------End of Global Run--------------------
 The run was 10000000 events
Graphics systems deleted.
Visualization Manager deleting...
File IAEAPSF/VarianClinaciX_6MV_4x4.IAEAphsp closed successfully!
```

14

Step 3-1: How to score dose depth curve

⑬ Score the dose in water phantom using built-in scorerer

```
#### verbosities
/control/verbose 1
/material/verbose 0            in run.mac
/run/verbose 0
/process/em/verbose 0

#/run/numberOfThreads 4

/physics/addPhysics standard_opt3

/PrimaryGenerator/useiaeaFile true                              # handson S2
/PrimaryGenerator/iaeaFileName IAEAPSF/VarianClinaciX_6MV_4x4   # handson S2
/PrimaryGenerator/gunOffset  0 cm                              # handson S2

/run/initialize

#/DetectorConstruction/PlateIonChamber/sensitiveMaterial G4_AIR  # handson S4
#/DetectorConstruction/PlateIonChamber/sensitiveMaterial G4_WATER # handson S4

/score/create/boxMesh boxMesh_1                     # handson S3
/score/mesh/translate/xyz 0 0 130 mm               # handson S3
/score/mesh/boxSize 127.5 127.5 127.5 mm  # half dim  # handson S3
/score/mesh/nBin 51 51 51                          # handson S3
/score/quantity/doseDeposit dDep                   # handson S3
/score/close                                       # handson S3
/score/list                                        # handson S3

#
#/run/printProgress 100000
/run/beamOn 10000000  # handson S1
#/run/beamOn 99999908 # handson S2 for IAEAPSF/VarianClinaciX_6MV_4x4

#/score/drawProjection boxMesh_1 dDep
/score/dumpQuantityToFile boxMesh_1 dDep dDep.txt # handson S3

~
~
~
~/Work/AIST/G4Work/temp/run.mac[+]
```
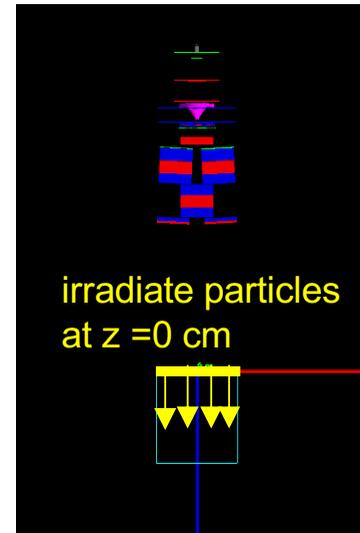
⑭ Execute the command

```
$ ./medical_linac run.mac
```

GEANT4-DNA

15

Step 3-2: How to score dose depth curve

Output (dDep.txt)

```
# mesh name: boxMesh_1
# primitive scorer name: dDep
# iX, iY, iZ, total(value) [Gy], total(val^2), entry
0,0,0,7.796356176683776e-13,3.569651199357745e-25,2
0,0,1,0,0,0
0,0,2,0,0,0
0,0,3,6.950653182524247e-13,4.831157966373445e-25,1
0,0,4,0,0,0
0,0,5,3.570276338500415e-13,1.274562847554965e-25,2
0,0,6,8.599334722933601e-14,7.269877901436412e-27,3
0,0,7,0,0,0
0,0,8,0,0,0
0,0,9,9.816770899260732e-14,9.636899088857236e-27,1
0,0,10,0,0,0
0,0,11,0,0,0
0,0,12,0,0,0
0,0,13,2.084636259308203e-13,4.345708333622496e-26,1
0,0,14,2.883633751315342e-13,8.315343611724991e-26,1
0,0,15,6.961137370037926e-16,4.845743348453854e-31,1
0,0,16,0,0,0
0,0,17,6.52242274348606e-14,4.254199844474423e-27,1
0,0,18,8.998883944111127e-13,6.99633712694326e-25,2
0,0,19,1.846215513903086e-12,2.189317572626159e-24,3
0,0,20,9.774618772686162e-13,7.443959118805567e-25,2
0,0,21,2.416287802079945e-13,3.50603901835526e-26,4
0,0,22,4.262672357678367e-13,5.813467009341347e-26,4
0,0,23,1.018564440357023e-13,7.452892332779504e-27,3
0,0,24,1.04208627532287e-13,6.641112973223942e-27,2
0,0,25,7.710901093569812e-14,5.937582280697706e-27,2
0,0,26,7.805803782701159e-13,3.917591186055819e-25,3
0,0,27,1.195141306212593e-12,1.152844398550952e-24,2
0,0,28,1.434759264955879e-12,7.563865324732861e-25,3
0,0,29,7.075212351823224e-17,3.146932977555055e-33,2
0,0,30,1.816851752161193e-13,3.300950289331196e-26,1
0,0,31,3.431623453588294e-13,6.7047690074913732e-26,2
0,0,32,3.1761541750012638e-13,5.667579638820665e-26,3
0,0,33,3.407273834474835e-13,1.057819255220334e-25,3
0,0,34,1.020617802811879e-12,5.603239193991691e-25,3
~/Work/AIST/G4Work/temp/dDep.txt
```

Analysis Code (Draw_PDD.C)

```cpp
struct DataEntry {
    int iX, iY, iZ;
    double totalValue;
    double totalValueSquared;
    int entry;
    };

void Draw_PDD(){

    ifstream file("dDep.txt");

    string line;
    vector<DataEntry> data;

    int nbins =51;
    TH1F *h1pdd = new TH1F("pdd","pdd",nbins,0,nbins);
    TH1F *h1ocr = new TH1F("ocr","ocr",nbins,0,nbins);

    while (std::getline(file, line)) {
        if (line.empty() || line[0] == '#') continue;
        stringstream ss(line);
        string token;
        DataEntry entry;

        getline(ss, token, ',');entry.iX = std::stoi(token);
        getline(ss, token, ',');entry.iY = std::stoi(token);
        getline(ss, token, ',');entry.iZ = std::stoi(token);
        getline(ss, token, ',');entry.totalValue = std::stod(token);
        getline(ss, token, ',');entry.totalValueSquared = std::stod(token);
        getline(ss, token, ',');entry.entry = std::stoi(token);

        if(entry.iX==25 && entry.iY==25){
            h1pdd->SetBinContent(entry.iZ,entry.totalValue);
        }
        if(entry.iY==25 && entry.iZ==5){
            h1ocr->SetBinContent(entry.iX,entry.totalValue);
        }

    }

    TCanvas *cpdd = new TCanvas("cpdd","cpdd",1200,400);
    cpdd->Divide(2,1);
    cpdd->cd(1);
    h1pdd->SetTitle("");
    h1pdd->SetXTitle("bin");
    h1pdd->SetYTitle("Normilized Dose");
    h1pdd->Scale(1./(double)h1pdd->GetMaximum());
    h1pdd->Draw("hist");
    cpdd->cd(2);
    h1ocr->SetTitle("");
    h1ocr->SetXTitle("bin");
    h1ocr->SetYTitle("Normilized Dose");
    h1ocr->Scale(1./(double)h1ocr->GetMaximum());
    h1ocr->Draw("hist");
}
```

⑮ Execute the command

$ root –l Draw_PDD.C

Step 3-3: Results
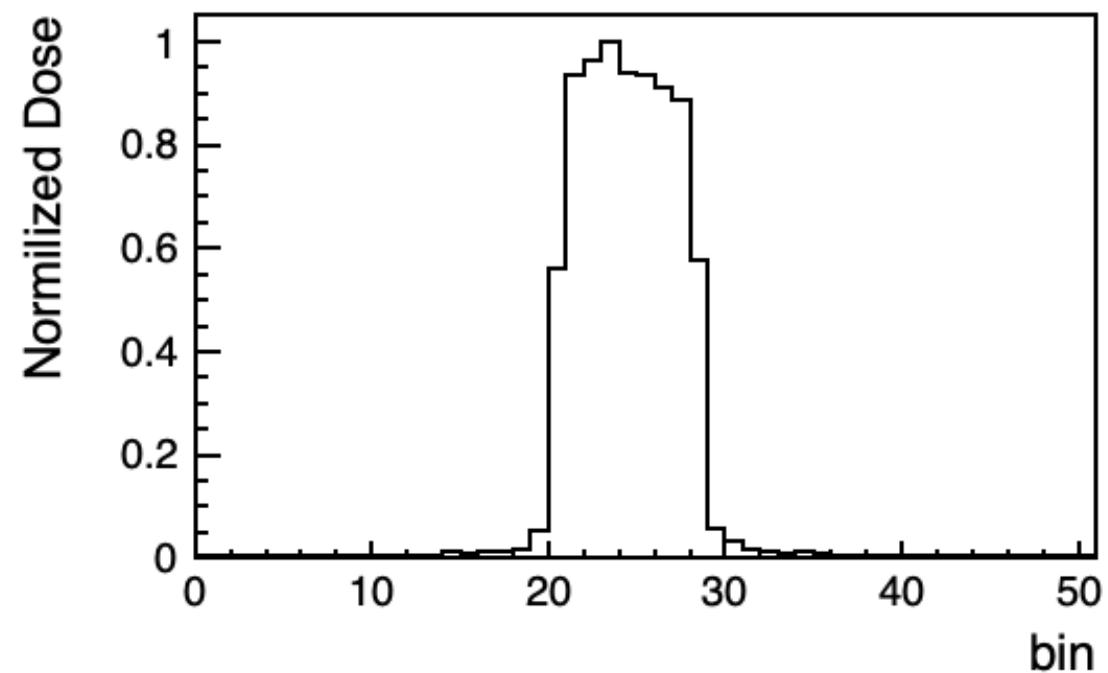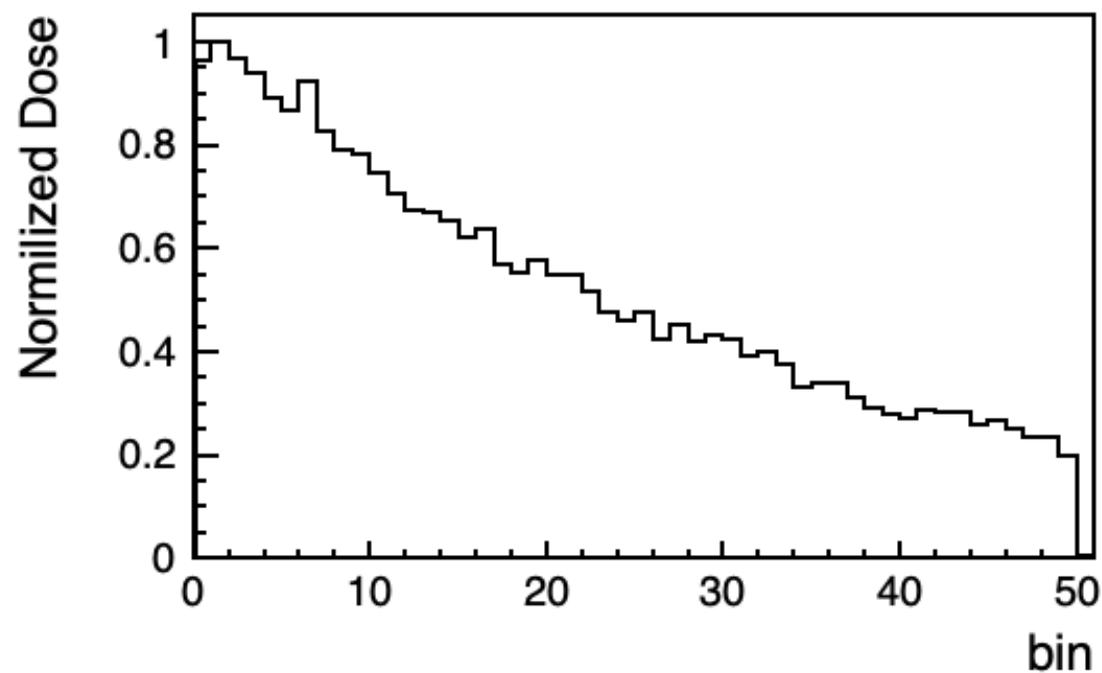
execute .q in the root command interface to exit

```
root [3] .q
g4vm:/build < 294 >
```

Hands-on completed!
Congratulations!

$\rightarrow$ Next page, advance hands-on
for those who are not novice users.
(We don't support this section in the tutorial)

Step 4-1: Dosimetry using Ionization Chamber (IC)
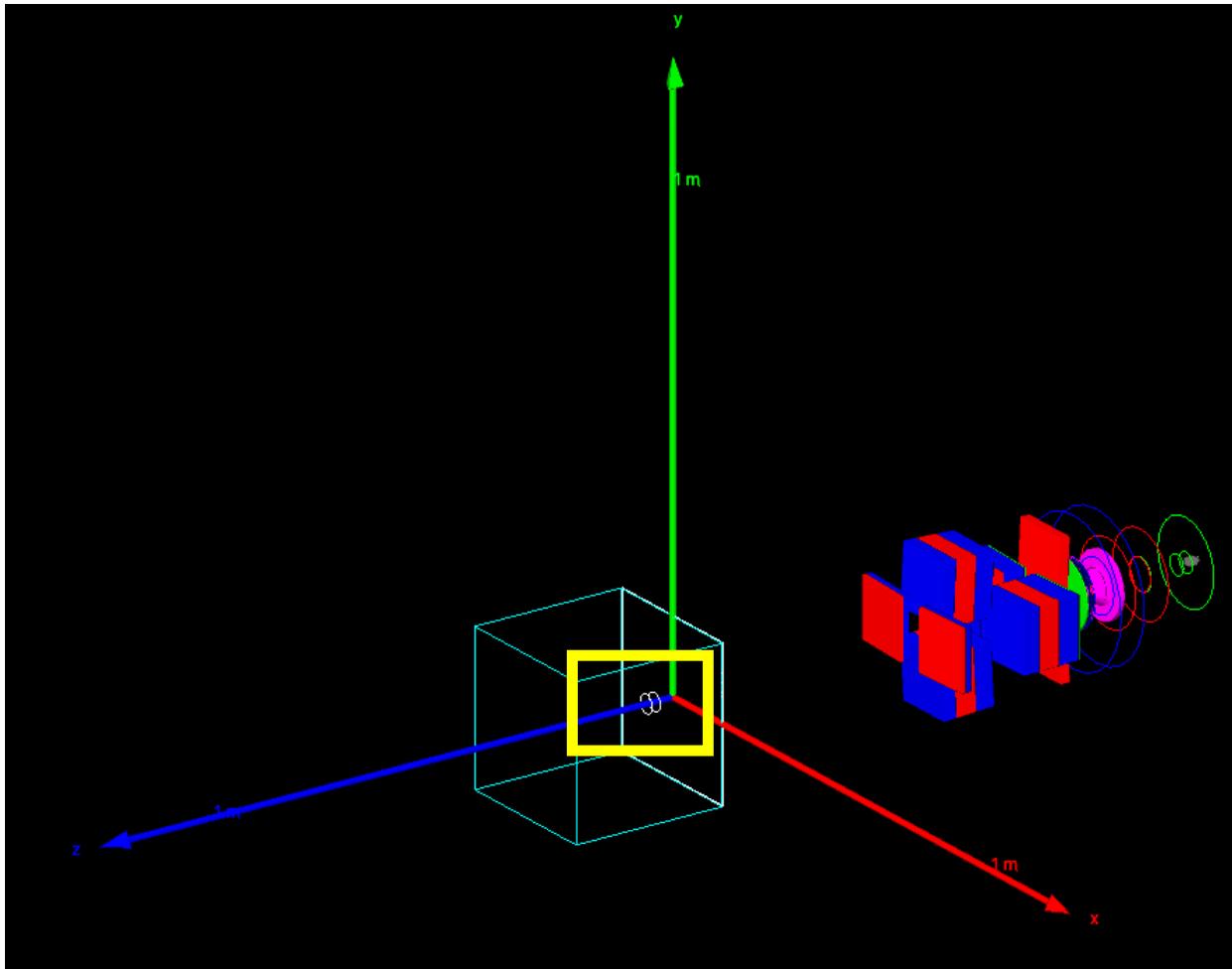
Score absorbed dose using ionization chamber

Plane parallel ionization chamber

1 mm   1 mm

5 mm

Sensitive volume

30 mm

14 mm
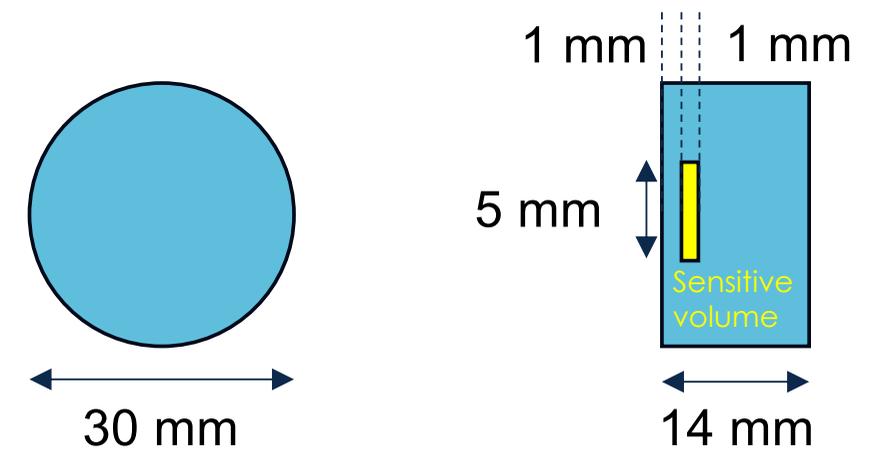
Place IC surface at 3 cm depth

Step 4-2: Construct an Ionization Chamber (IC)

DetectorConstruction.hh

```cpp
G4Material* mat_Kapton;
G4Material* mat_XC10;
G4Material* mat_WNICU;
G4Material* mat_Ssteel;
G4ThreeVector accHalfSize;
G4ThreeVector jaw1XInitialPos;
G4ThreeVector jaw2XInitialPos;
G4ThreeVector jaw1YInitialPos;
G4ThreeVector jaw2YInitialPos;
G4double jawAperture;
G4double fieldSide;
G4double sourceToSkinDistance;
G4double phantomSideDim, voxelSideDim, voxelDepthDim;
G4int nSideCells, nDepthCells;
G4double tubeFFRadius, tubeFFFirstFaceZ;
G4MultiFunctionalDetector* phantomDetector3D;
G4VPrimitiveScorer* phantomScorer3D;

G4Material* mat_icsens; // handson S4
G4VPhysicalVolume* ic_phys;// handson S4
};

#endif
```
`~/Work/AIST/G4Work/medical_linac_handson/include/DetectorConstruction.hh`

DetectorConstruction.cc

```cpp
#include <G4LogicalVolumeStore.hh>
#include <G4SolidStore.hh>
#include <G4Transform3D.hh>
#include <CLHEP/Vector/Rotation.h>

#include "G4SDManager.hh" //handson S4

G4VPhysicalVolume* DetectorConstruction::Construct()
{
 detectorMessenger = new DetectorMessenger(this);


 //--------- Visualization attributes ---------------------------
 /// Phantom
 G4VisAttributes* simpleH2OVisAtt= new G4VisAttributes(G4Colour::Cyan());
 simpleH2OVisAtt->SetVisibility(true);
 simpleH2OVisAtt->SetForceSolid(false);
 Phantom_log->SetVisAttributes(simpleH2OVisAtt);


 G4double icbodyradius = 15*mm;// handson S4
 G4double icbodythick  = 14*mm;// handson S4
 G4double icdepth = 3*cm;      // handson S4
 G4Tubs *icbody        = new G4Tubs("icbody", 0 , icbodyradius, icbodythick/2.,0,CLHEP::twopi);// handson S4
 //G4Material* pmma = G4NistManager::Instance()->FindOrBuildMaterial("G4_PLEXIGLASS");        // handson S4
 G4Material* grap = G4NistManager::Instance()->FindOrBuildMaterial("G4_GRAPHITE_POROUS");       // handson S4
 G4LogicalVolume* icbodyLV = new G4LogicalVolume(icbody, grap, "icbodyLV", 0, 0, 0);        // handson S4
 G4VPhysicalVolume* icbody_phys = new G4PVPlacement(nullptr, {0., 0., -phantomHalfDim.getZ()+icbodythick/2+icdepth}, "icbody", icbodyLV,  phantom_phys, false, 1);// handson S4

 G4double windowthick       = 1  *mm; // handson S4
 G4double sensitivevolumeradius = 2.5*mm; // handson S4
 G4double sensitivevolumethick  = 1  *mm; // handson S4
 G4Tubs *icsens        = new G4Tubs("icsens", 0, sensitivevolumeradius,sensitivevolumethick/2.,0.,CLHEP::twopi);// handson S4
 mat_icsens = G4NistManager::Instance()->FindOrBuildMaterial("G4_AIR");                // handson S4
 G4LogicalVolume* icsensLV = new G4LogicalVolume(icsens, mat_icsens, "icsensLV", 0, 0, 0);              // handson S4
 ic_phys = new G4PVPlacement(nullptr, {0., 0., -icbodythick/2. + sensitivevolumethick/2. + windowthick }, "icsens", icsensLV,  icbody_phys, false, 1);// handson S4

 auto plateionchamber = new G4MultiFunctionalDetector("plateionchamber"); // handson S4
 G4SDManager::GetSDMpointer()->AddNewDetector(plateionchamber);           // handson S4
 G4VPrimitiveScorer* primitiv = new G4PSDoseDeposit("dose");              // handson S4
 //see other classes G4PSFlatSurfaceCurrent, G4PSFlatSurfaceFlux          // handson S4
 plateionchamber->RegisterPrimitive(primitiv);                           // handson S4
 SetSensitiveDetector("icsensLV", plateionchamber);                      // handson S4

}
```
`~/Work/AIST/G4Work/medical_linac_handson/src/DetectorConstruction.cc`   `[Line:99/1105]`

Step 4-3: Check the geometry

```
#### verbosities
/control/verbose 1
/material/verbose 0
/run/verbose 0
/process/em/verbose 0

#/run/numberOfThreads 4

/physics/addPhysics standard_opt3

/PrimaryGenerator/useiaeaFile true                          # handson S2
/PrimaryGenerator/iaeaFileName IAEAPSF/VarianClinaciX_6MV_4x4    # handson S2
/PrimaryGenerator/gunOffset  0 cm                           # handson S2

/run/initialize

#/DetectorConstruction/PlateIonChamber/sensitiveMaterial G4_AIR  # handson S4
#/DetectorConstruction/PlateIonChamber/sensitiveMaterial G4_WATER # handson S4

#/score/create/boxMesh boxMesh_1                      # handson S3
#/score/mesh/translate/xyz 0 0 130 mm                 # handson S3
#/score/mesh/boxSize 127.5 127.5 127.5 mm  # half dim  # handson S3
#/score/mesh/nBin 51 51 51                            # handson S3
#/score/quantity/doseDeposit dDep                     # handson S3
#/score/close                                         # handson S3
#/score/list                                          # handson S3

#
#/run/printProgress 100000
/run/beamOn 10000000  # handson S1
#/run/beamOn 99999908 # handson S2 for IAEAPSF/VarianClinaciX_6MV_4x4

#/score/drawProjection boxMesh_1 dDep
#/score/dumpQuantityToFile boxMesh_1 dDep dDep.txt # handson S3

~
~
~
~/Work/AIST/G4Work/medical_linac_handson/run.mac[+]                    [Line:10/35]
```

```
$ make
$ ./medical_linac
```
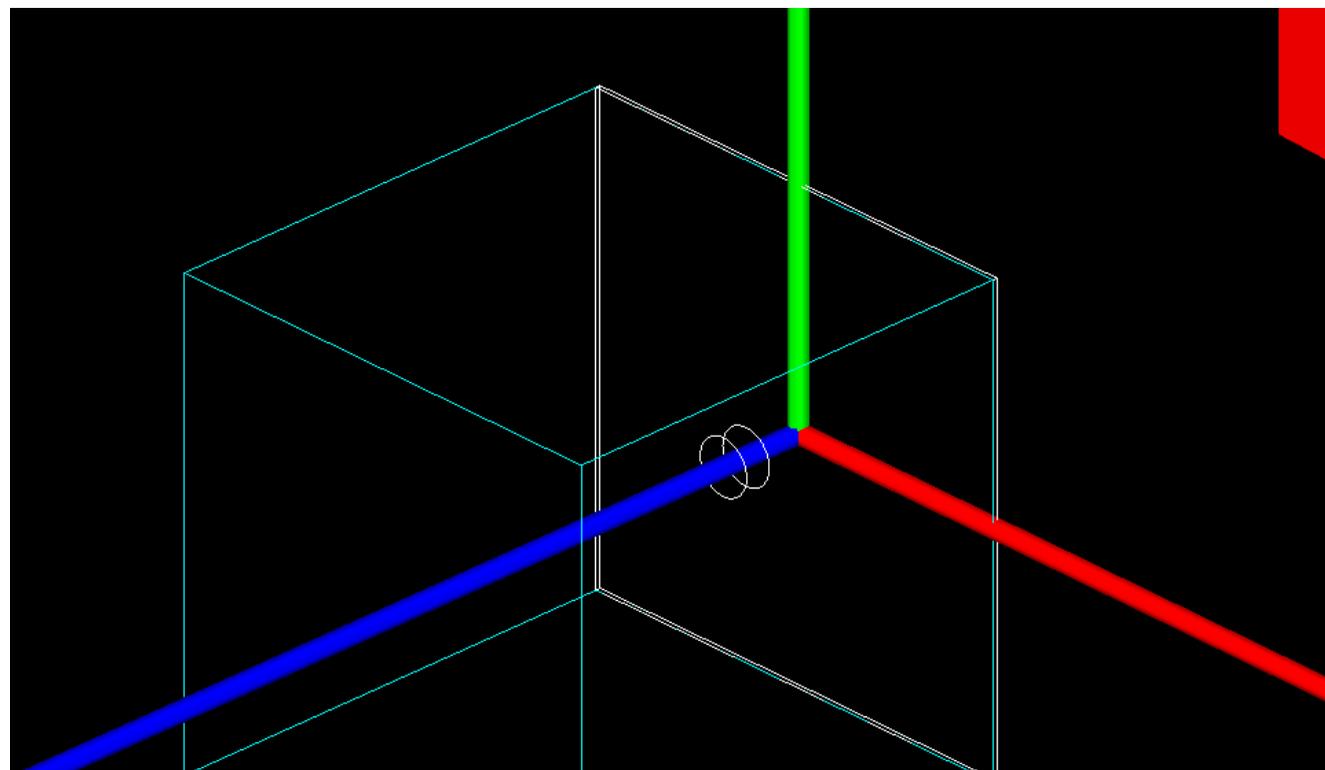
Step 4-3: Try to score the dose

RunAction.hh

```cpp
#ifndef RUNACTION_HH
#define RUNACTION_HH
#include <G4UserRunAction.hh>

class RunAction : public G4UserRunAction
{
public:          // handson S4
  RunAction();   // handson S4
  ~RunAction() ;// handson S4
public:
 void BeginOfRunAction(const G4Run* aRun) override; // handson S4
 void EndOfRunAction(const G4Run* aRun) override;   // handson S4
 //void EndOfRunAction(const G4Run*); //handson S4 comment out

};

#endif
~
```

> $ make
> $ ./medical_linac run.mac

```
... write file : output.root - done
... close file : output.root - done
... delete empty file : output.root - done

--------------------End of Global Run---------------------
 The run was 10000000 events
Graphics systems deleted.
Visualization Manager deleting...
File IAEAPSF/VarianClinaciX_6MV_4x4.IAEAphsp closed successfully!
```

RunAction.cc

```cpp
#include "RunAction.hh"
#include "G4Run.hh"

#include "G4AnalysisManager.hh"      //handson S4
#include "G4RootAnalysisManager.hh" //handson S4

RunAction::RunAction()                                      // handson S4
{                                                           // handson S4
  auto am = G4AnalysisManager::Instance();                  // handson S4
  am->SetFileName("output");                                // handson S4
  am->SetVerboseLevel(1);                                   // handson S4
  am->SetDefaultFileType("root");                           // handson S4
  am->SetNtupleDirectoryName("ntuple");                     // handson S4
  am->CreateNtuple("SimInfo", "Event by Event data");// handson S4
  am->CreateNtupleDColumn("dose");                          // handson S4
  am->FinishNtuple();                                       // handson S4
}                                                           // handson S4
RunAction::~RunAction() //handson S4
{                       //handson S4
}                       //handson S4
void RunAction::BeginOfRunAction(const G4Run* run)// handson S4
{                                                           // handson S4
  auto am = G4AnalysisManager::Instance();        // handson S4
  am->OpenFile();                                 // handson S4
}                                                 // handson S4
void RunAction::EndOfRunAction(const G4Run* run)
{
  auto am = G4AnalysisManager::Instance();        // handson S4
  am->Write();                                    // handson S4
  am->CloseFile();                                // handson S4

  G4int nofEvents = run->GetNumberOfEvent();
  if (IsMaster())
    {
      G4cout
      << "\n--------------------End of Global Run---------------------"
      << " \n The run was " << nofEvents << " events " << G4endl;
    }
}
```
`~/Work/AIST/G4Work/medical_linac_handson/src/RunAction.cc[+]`

but… the output file is always empty… why? : You did not read/save the dose!

GEANT4-DNA

Step 4-4: Score and Save the dose

So... How? → See EventAction_handson.cc

```cpp
void EventAction::EndOfEventAction(const G4Event* evt)
{
  auto am = G4AnalysisManager::Instance();

  // Hits collections
  G4HCofThisEvent* HCE = evt->GetHCofThisEvent();
  if (!HCE) return;

  if (fCollID_plateionchamber < 0) {
    G4SDManager* SDMan = G4SDManager::GetSDMpointer();
    fCollID_plateionchamber = SDMan->GetCollectionID("plateionchamber/dose");
  }

  auto evtMap = static_cast<G4THitsMap<G4double>*>(HCE->GetHC(fCollID_plateionchamber));

  for (auto& mapElement : (*evtMap->GetMap())) {
    auto dose = *(mapElement.second);
    auto copyNb = mapElement.first;
    //G4cout << " plate ion chamber :No-" << copyNb << ": " << dose/gray << " Gy " <<G4endl;
    am->FillNtupleDColumn(0, 0, dose/gray);
    am->AddNtupleRow();

  }

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

`~/Work/AIST/G4Work/medical_linac_handson/src/EventAction_handson.cc`

```
$ make
$ ./medical_linac run.mac
```

Then, output.root is created!

And activate it via ActionInitialization.cc

```cpp
#include "ActionInitialization.hh"
#include "PrimaryGeneratorAction.hh"
#include "RunAction.hh"
#include "EventAction_handson.hh"//handson S4
#include "StackingAction.hh"
#include "SteppingAction.hh"

void ActionInitialization::Build() const
{
  SetUserAction(new PrimaryGeneratorAction());
  RunAction* theRunAction = new RunAction();
  SetUserAction(theRunAction);
  SetUserAction(new EventAction(theRunAction)); //handson S4
  SetUserAction(new StackingAction());
  SetUserAction(new SteppingAction());
}

void ActionInitialization::BuildForMaster() const
{
  SetUserAction(new RunAction());
}
~
~
~
~
```

`~/Work/AIST/G4Work/medical_linac_handson/src/ActionInitialization.cc[+]`

Analysis code (Calc_dose.C)

```cpp
void Calc_dose(){
    TFile* f = new TFile("output.root", "READ");
    //f->ls();

    TTree* tree = (TTree*)f->Get("ntuple/SimInfo");
    //tree->Print();

    double dose=0;
    double totaldose=0;
    double totaldose_e=0;

    vector<double> vdose;

    int entries = tree->GetEntries();

    tree->SetBranchAddress("dose", &dose);
    for(int i=0; i<entries; ++i){
        tree->GetEntry(i);
        vdose.push_back(dose);
    }

    totaldose = calculateSum(vdose);
    totaldose_e = calculateStandardError(vdose);

    cout<< " Total Dose = " << totaldose <<" +- "<< totaldose_e<<" Gy "<<endl;


}
```

$ root –l Calc_dose.C

Step 4-6: Set function to change the material of sensitive volume

DetectorConstruction.hh

```cpp
class G4LogicalVolume;
class DetectorMessenger;
class DetectorConstruction : public G4VUserDetectorConstruction
{
public:

 G4VPhysicalVolume* Construct() override;

 void SetJaws(G4double value);
 void SetTargetPosition(G4double value);
 void SetPhantomSide(G4double value);
 void SetVoxelSide(G4double value);
 void SetVoxelDepth(G4double value);
 void SetICSensMaterial(G4String value);// handson S4

 void UpdateGeometry(G4String, G4double);
 void UpdateGeometry(G4String, G4String);// handson S4

 G4int GetNumberSideCells() const;
 G4int GetNumberDepthCells() const;
 G4int GetPhantomDepth() const;
```
`~/Work/AIST/G4Work/medical_linac_handson/include/DetectorConstruction.hh`

DetectorConstruction.cc

```cpp
void DetectorConstruction::SetVoxelDepth(G4double value)
{
 voxelDepthDim = value;
}
void DetectorConstruction::SetICSensMaterial(G4String value)        // handson S4
{                                                                   // handson S4
  mat_icsens = G4NistManager::Instance()->FindOrBuildMaterial(value);// handson S4
}                                                                   // handson S4


 else
  G4cerr << "*** DetectorMessenger::UpdateGeometry: Command not found" << G4endl;

G4RunManager::GetRunManager()->GeometryHasBeenModified();
}
void DetectorConstruction::UpdateGeometry(G4String string, G4String value)     // handson S4
{                                                                              // handson S4
 if (string == "icsensMat")                                                    // handson S4
 {                                                                             // handson S4
   ic_phys -> GetLogicalVolume() ->SetMaterial(G4NistManager::Instance()->FindOrBuildMaterial(value));// handson S4
 }                                                                            // handson S4
 else                                                                         // handson S4
  G4cerr << "*** DetectorMessenger::UpdateGeometry: Command not found" << G4endl;// handson S4

G4RunManager::GetRunManager()->GeometryHasBeenModified();                      // handson S4
}                                                                              // handson S4
```
`~/Work/AIST/G4Work/medical_linac_handson/src/DetectorConstruction.cc`

GEANT4-DNA

Step 4-7: Set messenger class

DetectorMessenger.hh

```cpp
#ifndef DETECTOR_MESSENGER_HH
#define DETECTOR_MESSENGER_HH

#include <G4UImessenger.hh>
#include <G4UIdirectory.hh>
#include <G4UIcmdWithAString.hh>// handson S4
#include <G4UIcmdWithADouble.hh>
#include <G4UIcmdWithoutParameter.hh>
#include <G4UIcmdWithADoubleAndUnit.hh>
#include <G4SystemOfUnits.hh>

#include "DetectorConstruction.hh"

class DetectorConstruction;
class G4UIdirectory;
class G4UIcmdWithADoubleAndUnit;

/// Messenger class that defines commands for DetectorConstruction.
///
/// It implements commands:
/// – Acc/fieldSide
/// – Acc/sourceToSkinDistance
/// – Phantom/phantomSide
/// – Phantom/voxelSide
/// – Phantom/voxelDepth

class DetectorMessenger : public G4UImessenger
{
public:
 DetectorMessenger(DetectorConstruction* );
 ~DetectorMessenger();
 void SetNewValue(G4UIcommand*, G4String) override;

private:
 DetectorConstruction* detPointer;

 G4UIcmdWithADoubleAndUnit* fieldSide;
 G4UIcmdWithADoubleAndUnit* sourceToSkinDistance;
 G4UIcmdWithADoubleAndUnit* phantomSide;
 G4UIcmdWithADoubleAndUnit* voxelSide;
 G4UIcmdWithADoubleAndUnit* voxelDepth;
 G4UIcmdWithAString* icsensMat;// handson S4
};

#endif
~
~
~
~
```
`~/Work/AIST/G4Work/medical_linac_handson/include/DetectorMessenger.hh[+]`

DetectorMessenger.cc

```cpp
#include "DetectorMessenger.hh"
#include <G4RunManager.hh> //handson S4


DetectorMessenger::~DetectorMessenger()
{
 // UI commands and directory have to be deleted
 delete fieldSide;
 delete sourceToSkinDistance;
 delete phantomSide;
 delete voxelSide;
 delete voxelDepth;
 delete icsensMat;//handson S4
}

 icsensMat = new G4UIcmdWithAString("/DetectorConstruction/PlateIonChamber/sensitiveMaterial", this); // handson S4
 icsensMat  -> SetDefaultValue("G4_AIR");                            // handson S4
 icsensMat  -> SetGuidance("Set the Material of SensitiveVolume"); // handson S4
 detPointer -> SetICSensMaterial("G4_AIR");                          // handson S4


 else if (cmd == icsensMat)                              // handson S4
 {                                                       // handson S4
  detPointer -> SetICSensMaterial(newValue);             // handson S4
  detPointer -> UpdateGeometry("icsensMat", newValue);   // handson S4
 }                                                       // handson S4
 else
  G4cerr << "DetectorMessenger::SetNewValue: command not found" << G4endl;
}
```
`~/Work/AIST/G4Work/medical_linac_handson/src/DetectorMessenger.cc`

Step 4-8: change the material of the sensitive volume

```
#### verbosities
/control/verbose 1
/material/verbose 0
/run/verbose 0
/process/em/verbose 0

#/run/numberOfThreads 4

/physics/addPhysics standard_opt3

/PrimaryGenerator/useiaeaFile true                          # handson S2
/PrimaryGenerator/iaeaFileName IAEAPSF/VarianClinaciX_6MV_4x4   # handson S2
/PrimaryGenerator/gunOffset  0 cm                           # handson S2

/run/initialize

#/DetectorConstruction/PlateIonChamber/sensitiveMaterial G4_AIR  # handson S4
/DetectorConstruction/PlateIonChamber/sensitiveMaterial G4_WATER # handson S4

#/score/create/boxMesh boxMesh_1                    # handson S3
#/score/mesh/translate/xyz 0 0 130 mm              # handson S3
#/score/mesh/boxSize 127.5 127.5 127.5 mm  # half dim  # handson S3
#/score/mesh/nBin 51 51 51                         # handson S3
#/score/quantity/doseDeposit dDep                  # handson S3
#/score/close                                      # handson S3
#/score/list                                       # handson S3

#
#/run/printProgress 100000
/run/beamOn 10000000  # handson S1
#/run/beamOn 99999908 # handson S2 for IAEAPSF/VarianClinaciX_6MV_4x4

#/score/drawProjection boxMesh_1 dDep
#/score/dumpQuantityToFile boxMesh_1 dDep dDep.txt # handson S3
```

```
$ make
$ ./medical_linac run.mac
$ root –l Calc_dose.C
```

Try to calculate the ratio of
   dose to water and dose to air !

Challenge! (without any help)
1. Try to implement macro command to
    change depth of the IC
2. Try to implement macro command to
    change material of IC-body