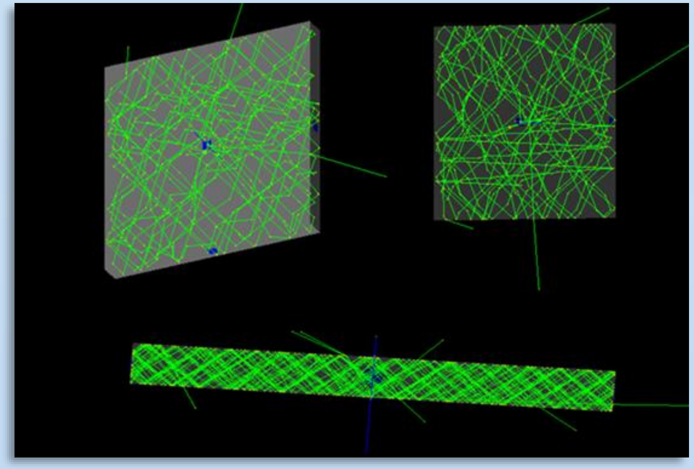
Simulation of optical photon propagation for generic scintillator-based detectors

Lecture 3

Introduction and Geometry



Davide Serini
davide.serini@ba.infn.it

Hands-on schedule

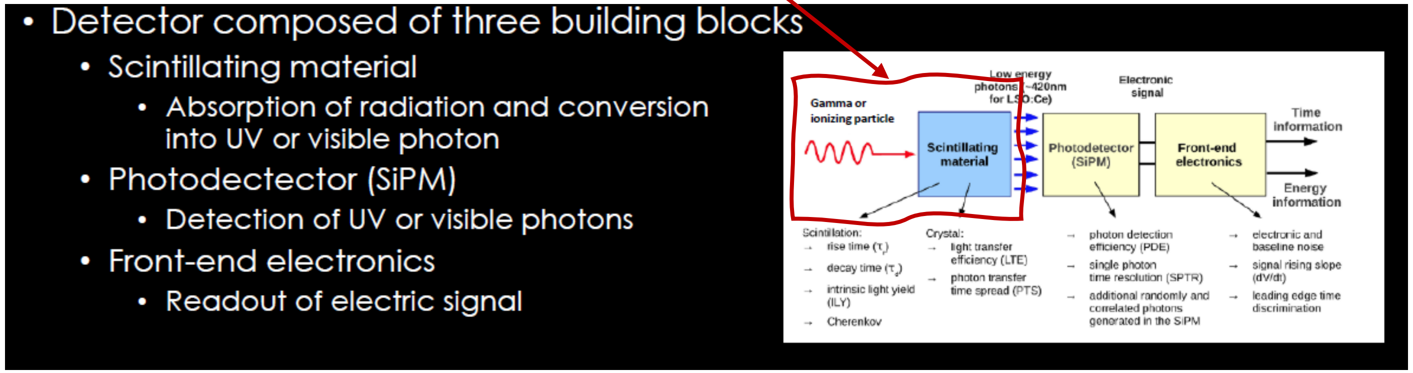
- General G4 introduction
  - How to run the example B1
  - Build and run the code
    - *Run a simple simulation using the User Interface*
    - *Run a simple simulation using a macro*

- Make your own project
  - Build your basic project:
    - *main & physics list*
    - *geometry*
    - *write a macro*
  - Build your simulation: data management tools:
    - *Sensitive Detector*
    - *Hit collection*
    - *Run Action*
    - *Event Action*
  - Make your simulation (final exercise)

General G4 introduction

- Some information about G4 (GEometry ANd Traking)
  - C++ Language
  - Open Source
  - It is a toolkit

- Composing a G4 application:
  - General Classes:
    - *Main() file*
    - *Sources files(*.cc)*
      - *Usually included in the src/ folder*
    - *Header files(*.hh)*
      - *Usually included in the include/ folder*
    - *Three classes are mandatory*
      - *PrimaryGenerationAction (.cc and .hh)*
      - *DetectorConstruction (.cc and .hh)*
      - *PhysicsList (.cc and .hh)*



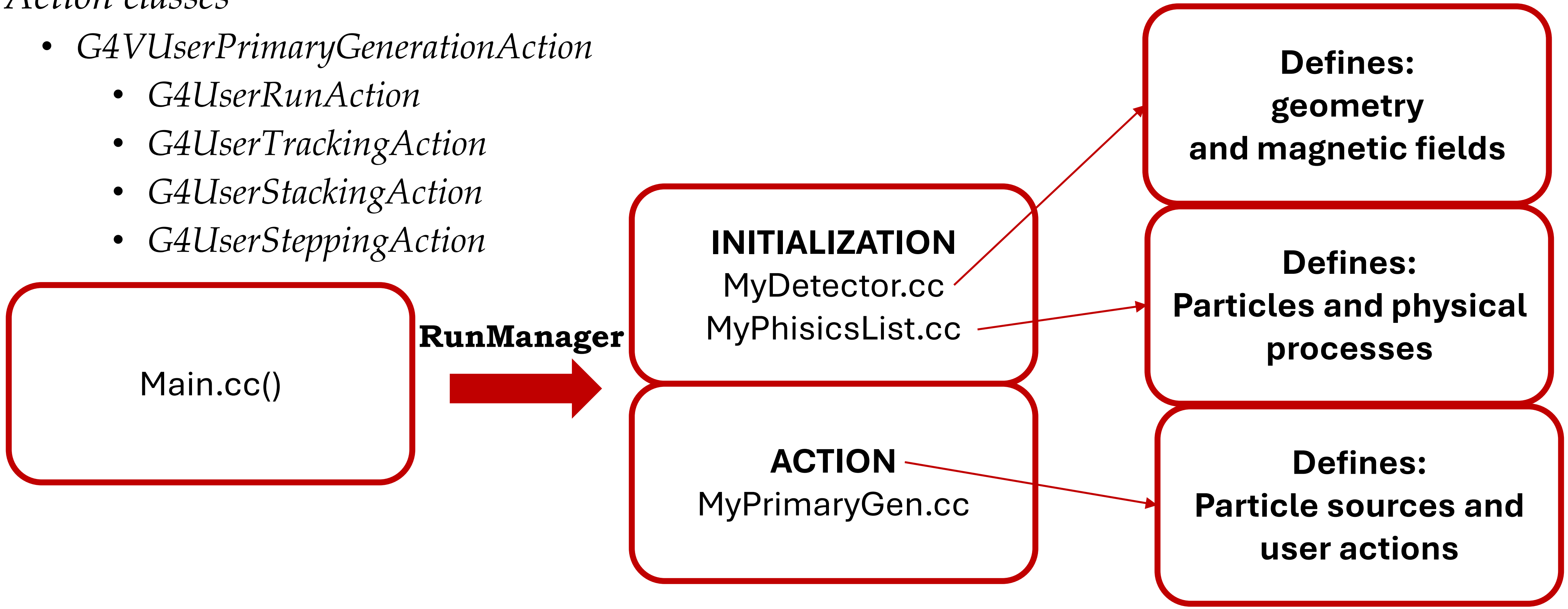
- Detector composed of three building blocks
  - Scintillating material
    - Absorption of radiation and conversion into UV or visible photon
  - Photodetector (SiPM)
    - Detection of UV or visible photons
  - Front-end electronics
    - Readout of electric signal

Gamma or ionizing particle
Low energy photons (~420nm for LSO:Ce)
Electronic signal
Scintillating material
Photodetector (SiPM)
Front-end electronics
Time information
Energy information
Scintillation:
→ rise time (τr)
→ decay time (τd)
→ intrinsic light yield (ILY)
→ Cherenkov
Crystal:
→ light transfer efficiency (LTE)
→ photon transfer time spread (PTS)
→ photon detection efficiency (PDE)
→ single photon time resolution (SPTR)
→ additional randomly and correlated photons generated in the SiPM
→ electronic and baseline noise
→ signal rising slope (dV/dt)
→ leading edge time discrimination

You must provide the necessary information to configure your simulation

General G4 introduction

- Composing a G4 application:
  - General Classes:
    - *Initialization classes*
      - *G4VUserDetectorConstrucion*
      - *G4VUserPhysicsList*
    - *Action classes*
      - *G4VUserPrimaryGenerationAction*
        - *G4UserRunAction*
        - *G4UserTrackingAction*
        - *G4UserStackingAction*
        - *G4UserSteppingAction*



General G4 introduction

- What is necessary to built a complete simulation:
  - Detector Construction:
    - *Build your geometry, define your materials and eventually assign optical bulk and surface properties*
  - Physics List:
    - *"Activate" the physical processes that can occur during your simulation*
  - Hit collection:
    - *Define your "hit container"; i.e. which are the information that you want store during your simulation*
  - Sensitive Detector:
    - *Define your definition of hit; i.e. when you want to store the information*
  - Action classes:
    - *G4UserRunAction:*
      - *Define what happen at the begin and at the end of the* ***complete simulation***
        - *Create an output file with its structure (Beginning)*
        - *Close the output file (end)*
    - *G4UserEventAction:*
      - *Define what happen at the begin and at the end of* ***each event***
        - *Retrieve the information of each hit*
        - *Fill the output*

Main function

- Geant4 does not provide a standard main() function
- The main() is part of the user application
- In the main() the user **must**
    - instantiate G4RunManager(or his/her own derived class)
    - notify the G4RunManager mandatory user classes derived from
        - *G4VUserDetectorConstruction*
        - *G4VUserPhysicsList*
        - *G4VUserPrimaryGeneratorAction*
- The user may also instantiate in the main() function
    - optional user action classes
    - visualisation manager, UI session

Main function

main()

```
{
 …
 // Instantiate the default run manager
 G4RunManager* runManager = new G4RunManager;
// Instantiate mandatory user initialization classes and notify
runManager
 MyDetectorConstruction* detector = new MyDetectorConstruction;
 runManager->SetUserInitialization(detector);
 MyPhysicsList* physicsList = new MyPhysicsList;
 runManager->SetUserInitialization(myPhysicsList);

 // Instantiate mandatory user action classes and notify runManager
 runManager->SetUserAction(new MyPrimaryGeneratorAction);

 // Instantiate optional user action classes and notify runMana
 MyEventAction* eventAction = new MyEventAction();
  runManager->SetUserAction(eventAction);
 MyRunAction* runAction = new MyRunAction();
  runManager->SetUserAction(runAction);
 …
```

A **run** is a collection of **events** that share the *same detector conditions*

- Detector and physics settings are frozen in a run

An **event** initially contains the *primary particles*; they are pushed into a stack and further processed

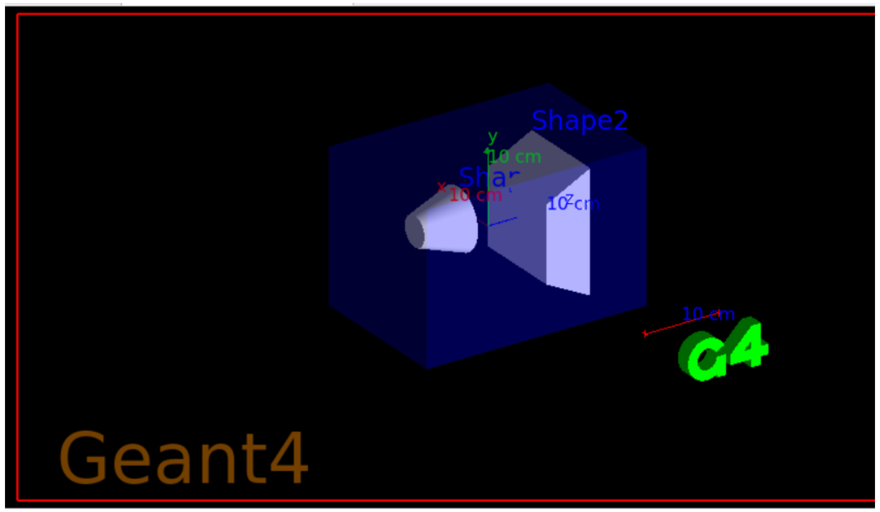
===

Geant4 - an Object-Oriented Toolkit for Simulation in HEP

===

Example B1

This example demonstrates a very simple application where an energy deposit is accounted in user actions and a dose in a selected volume is calculated.

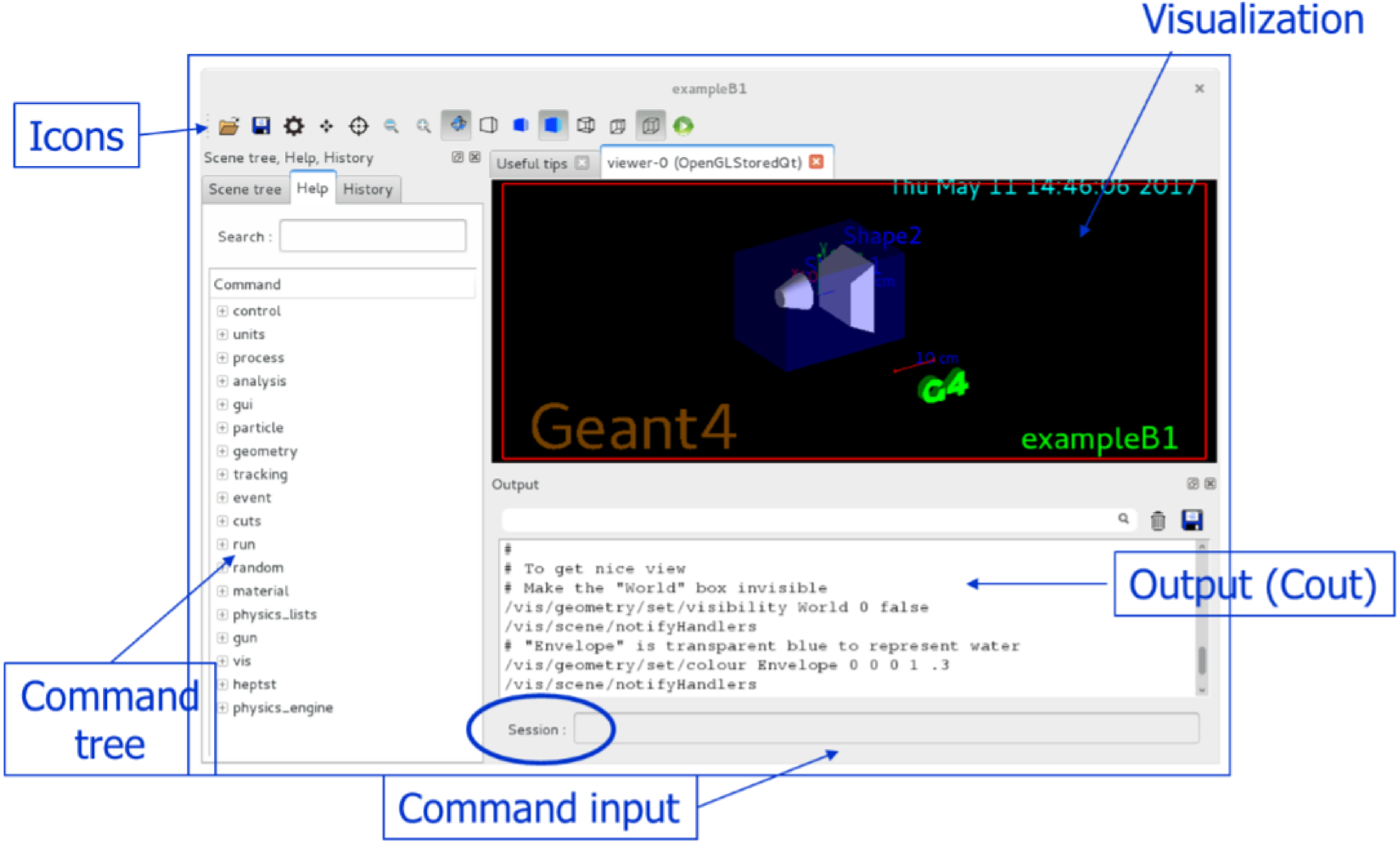


1- GEOMETRY DEFINITION

The geometry is constructed in the B1DetectorConstruction class. The setup consists of a an envelope of box shape containing two volumes: a spherical cone and a trapezoid.

In this example we use some common materials materials for medical applications. The envelope is made of water and the two inner volumes are made from tissue and bone materials.
The materials are created with the help of the G4NistManager class, which allows to build a material from the NIST database using their names. All available materials can be found in the Geant4 User's Guide for Application Developers, Appendix 10: Geant4 Materials Database.

User interface (Qt and OpenGL)



Visualization

Icons

Command tree

Output (Cout)

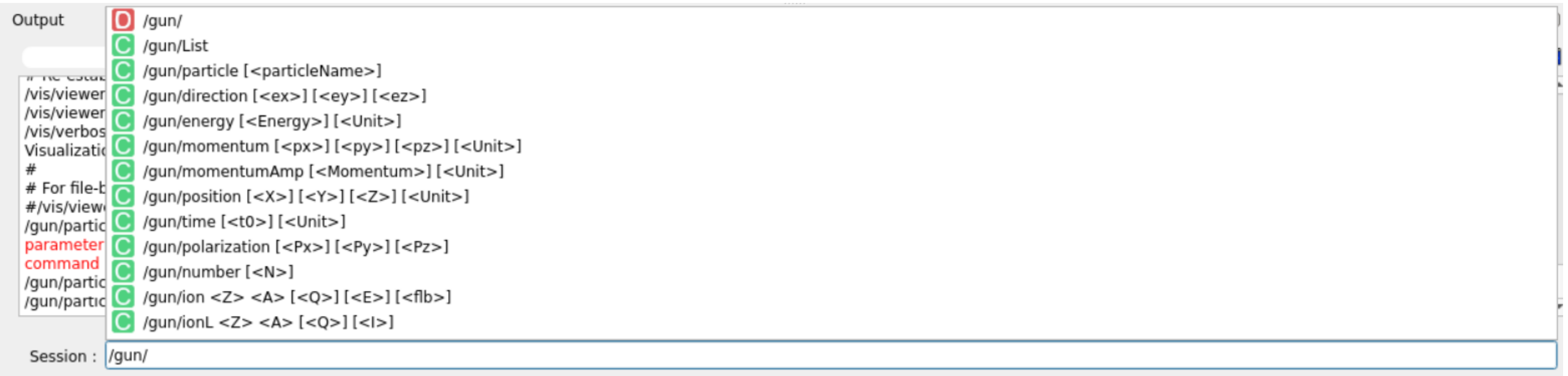
Command input

```
#
# To get nice view
# Make the "World" box invisible
/vis/geometry/set/visibility World 0 false
/vis/scene/notifyHandlers
# "Envelope" is transparent blue to represent water
/vis/geometry/set/colour Envelope 0 0 0 1 .3
/vis/scene/notifyHandlers
```

Explore Geant4 and B1 example

- How to compile a simulation
  - Create a directory ProjectName_build and compile your source code inside the directory
    - `make:`
      - `cmake /Path\to\Project_source (Example: `**`cmake /home/dserini/geant4/B1`**`)`
      - `Compile your source code with the command make`
      - Example: >> cd $HOME/B1_build
        >> cmake `/home/dserini/geant4/B1`
        >> ./exampleB1
    - Run the executable: `./exampleB1`
- Run a simulation using the Geant4 GUI:
  - G4ParticleGun (GUN) main commands (http://www.hep.ph.ic.ac.uk/~yoshiu/COMET/comet_g4HTMLdoc/_gun_.html)

```
/gun/particle proton
/gun/position 0 0 10 cm
/gun/direction 0 0 1
/gun/energy 1 GeV
/run/beamOn 10
```



```
/vis/viewer
/vis/viewer
/vis/verbos
Visualizatic
#
# For file-b
#/vis/viewe
/gun/partic
parameter
command
/gun/partic
/gun/partic
```

```
D /gun/
C /gun/List
C /gun/particle [<particleName>]
C /gun/direction [<ex>] [<ey>] [<ez>]
C /gun/energy [<Energy>] [<Unit>]
C /gun/momentum [<px>] [<py>] [<pz>] [<Unit>]
C /gun/momentumAmp [<Momentum>] [<Unit>]
C /gun/position [<X>] [<Y>] [<Z>] [<Unit>]
C /gun/time [<t0>] [<Unit>]
C /gun/polarization [<Px>] [<Py>] [<Pz>]
C /gun/number [<N>]
C /gun/ion <Z> <A> [<Q>] [<E>] [<flb>]
C /gun/ionL <Z> <A> [<Q>] [<I>]
```

Session : /gun/

  - G4GeneralParticleSource (GPS) main commands (http://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/ForApplicationDeveloper/html/GettingStarted/generalParticleSource.html):

User Interface (Qt and OpenGL)

- Some UI commands:
  - /run/verbose 1
    - *Sets how much output the run manager will print*
  - /run/initialize
    - *Initializes the run*
  - /run/beamOn 100
    - *Starts a run with 100 events*
  - /control/execute macroName.mac
    - *Run the commands in a macro file*

- A complete list of built-in command is available in the Application Developers Guide, Chapter 7

Links

Blank

```
wget 'https://istnazfisnucl-my.sharepoint.com/:u:/g/personal/serini_infn_it/EX-1XCKrJdlImqv1wKKZY20BiiMb9b5eAdP1ozxKD2wLyQ?e=Y7DJeZ&download=1' -O Blank.tar.gz
```