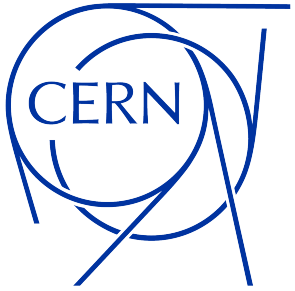
First Steps in Geant4: Examples and Documentation

Geant4–11.1 reference - Based on previous Geant4 courses

Lorenzo Pezzotti
CERN EP-SFT

10th International Geant4 Course in Korea 2023
@ Jeju National University



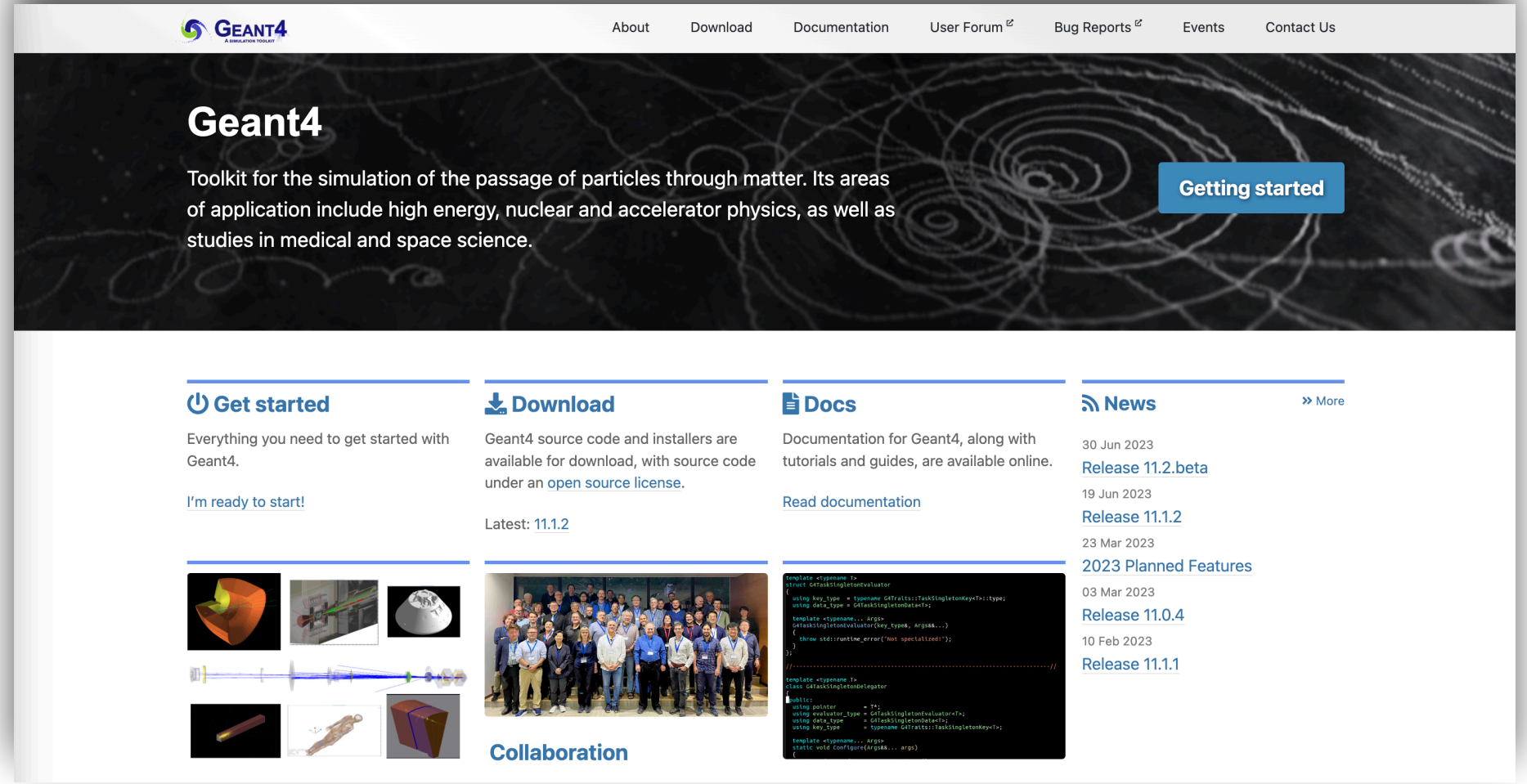
Covered topics

This lesson is about taking the first steps in `Geant4`, it shows how to sniff around *examples and documentation*

- **`Geant4` installation**
- **Examples**
  - Build and examine `exampleB1`
  - Basic examples
  - Extended examples
  - Advanced examples
- **Documentation**
  - Installation Guide
  - Documentation for Application/Toolkit Developers
- **User support**

Step #1: Geant4 website

- Where to find our documentation, download the code, access the user forum, inspect future events, …



Step #2: Install Geant4

The installation guide is available at https://geant4-userdoc.web.cern.ch/UsersGuides/InstallationGuide/html/

- Refer to the guide for installation *dependencies*, *supported platforms* and *cmake options*

Example of `Geant4` installation:

- Download source code from https://geant4.web.cern.ch/support/download or from github, then

Bash - Example of Geant4 (latest patch) installation

```
$ unzip geant4-11.1.2.zip

$ mkdir geant4-11.1.2-build; cd geant4-11.1.2-build

$ cmake -DCMAKE_INSTALL_PREFIX=/path-to-install/geant4-11.1.2-install
-DGEANT4_INSTALL_DATA=ON -DGEANT4_USE_QT=ON -DGEANT4_BUILD_MULTITHREADED=ON
/path-to/geant4-11.1.2/

$ make -j6

$ make install
```

Selected *cmake* options:
- download and install data libraries (see lessons on physics)
- use `qt` for visualization
- build `geant4` with multithreaded capability

Step #3: Build examples (1/2)

Examples (`geant4/examples/`) are useful applications to learn `Geant4` features (from basic to advanced)

- Building example `B1`, the first basic example

Bash - Building example B1

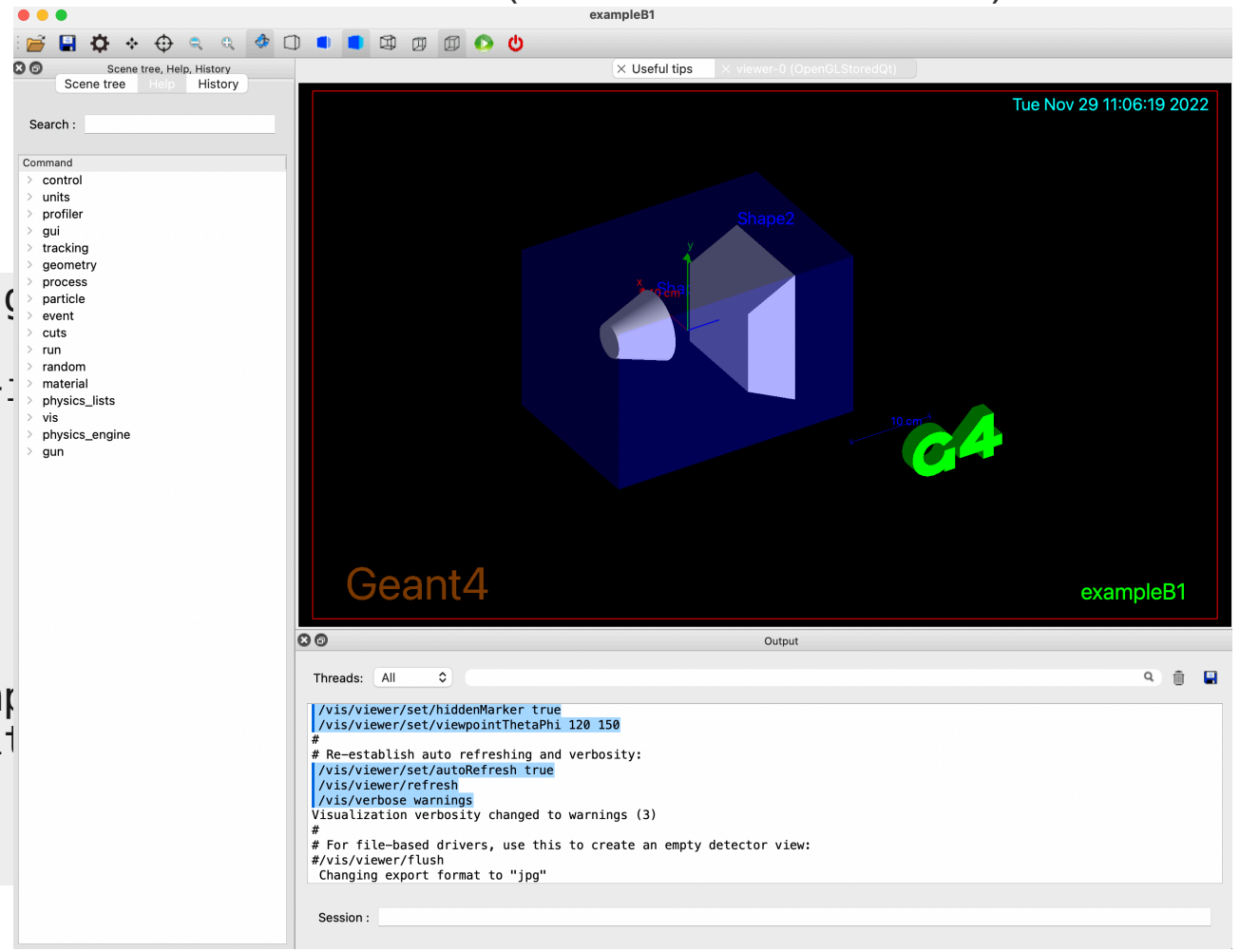
```
$ source /path-to/geant4-11.1.2-install/bin/geant4.sh

$ cmake -DGeant4_DIR=/path-to/geant4-11.1.2-install/lib/Geant4-11.1.2/
/path-to/geant4-11.1.2/examples/basic/B1/

$ make

$ ls

CMakeCache.txt   Makefile   exampleB1   exampleB1.out   run1.mac  vis.mac      CMakeFiles
cmake_install.cmake     exampleB1.in     init_vis.mac   run2.mac

$ ./exampleB1
```

Several files are created when you compile an example.
Look for the executable file and the *.mac macro card setting some parameters.

Step #3: Build examples (2/2)

Examples (`geant4/examples/`) are useful applications to learn `Geant4` features (from basic to advanced)

- Building example `B1`, the first basic example



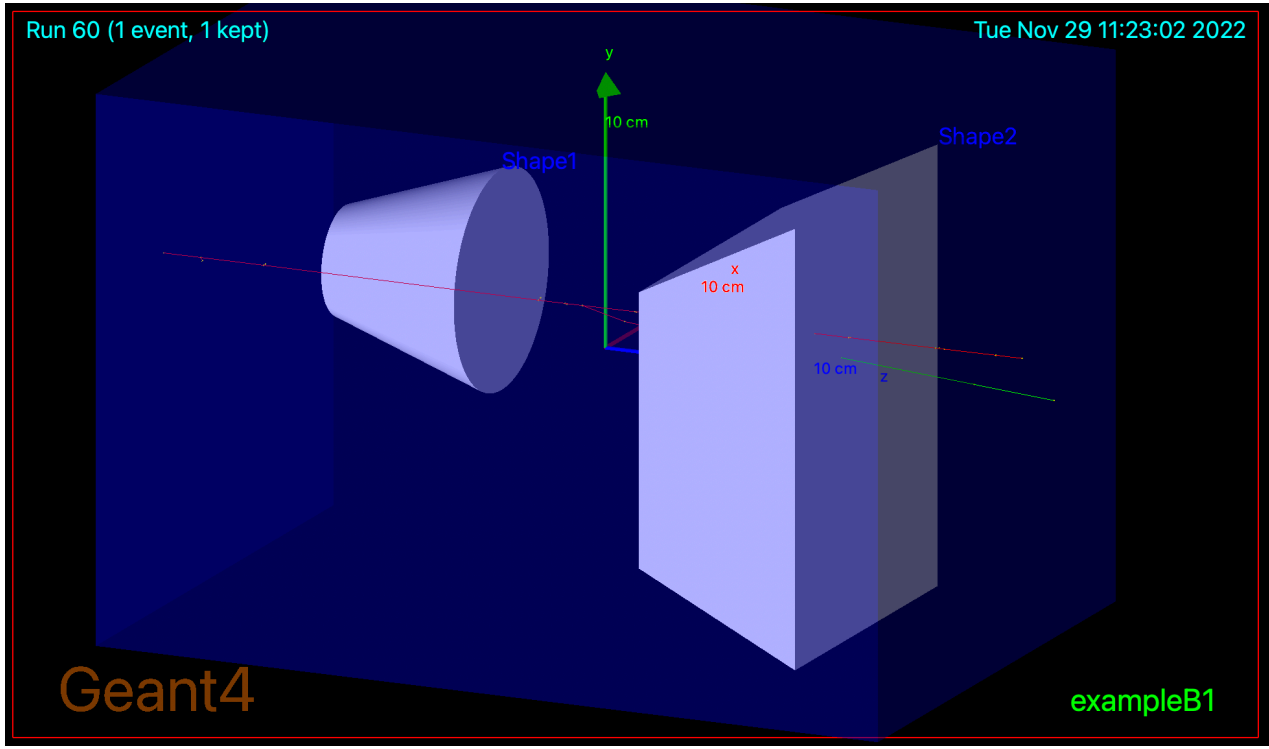
Bash - Building example B1

```
$ source /path-to/geant4-11.1.2-install/bin/g

$ cmake -DGeant4_DIR=/path-to/geant4-11.1.2-i
/path-to/geant4-11.1.2/examples/basic/B1/

$ make

$ ls

CMakeCache.txt   Makefile   exampleB1   examp
cmake_install.cmake     exampleB1.in     init

$ ./exampleB1
```

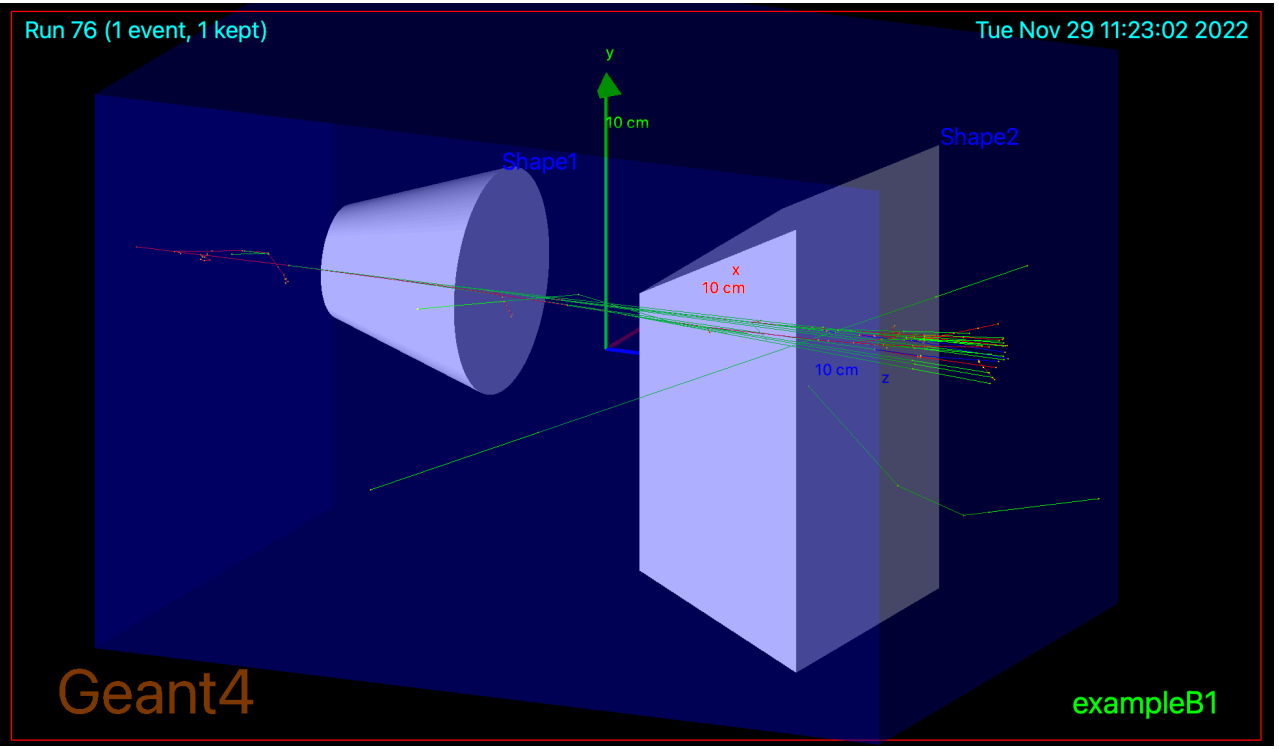
Step #4: Shoot particles

Once you have built any `Geant4` application, shooting particles inside and watching the result is *irresistible*

```
/gun/particle mu-
/gun/energy 1.0 GeV
/run/beamOn 1
```




```
/gun/particle e-
/gun/energy 1.0 GeV
/run/beamOn 1
```



Step #5: Sniff around example B1 (1/4)

Examining code from examples is one of the best ways to learn

- Example: the `main()` function

geant4-11.1.2/examples/basic/B1/exampleB1.cc

```
int main(int argc,char** argv) {
  // some code

  // Construct the default run manager
  //
  auto* runManager =
    G4RunManagerFactory::CreateRunManager(G4RunManagerType::Default);

  // Set mandatory initialization classes
  //
  // Detector construction
  runManager->SetUserInitialization(new DetectorConstruction());

  // Physics list
  G4VModularPhysicsList* physicsList = new QBBC;
  physicsList->SetVerboseLevel(1);
  runManager->SetUserInitialization(physicsList);

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

  // some code
}
```

Step #5: Sniff around example B1 (2/4)

Examining code from examples is one of the best ways to learn

- Example: the `main()` function

Every Geant4 application has a *run manager*

There are different run manager *types* (single-threaded *vs.* multi-threaded)

Both the *physic list* and the *detector geometry* are passed to the run manager

and *user actions* too…

geant4-11.1.2/examples/basic/B1/exampleB1.cc

```
int main(int argc,char** argv) {
  // some code

  // Construct the default run manager
  //
  auto* runManager =
    G4RunManagerFactory::CreateRunManager(G4RunManagerType::Default);

  // Set mandatory initialization classes
  //
  // Detector construction
  runManager->SetUserInitialization(new DetectorConstruction());

  // Physics list
  G4VModularPhysicsList* physicsList = new QBBC;
  physicsList->SetVerboseLevel(1);
  runManager->SetUserInitialization(physicsList);

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

  // some code
}
```

Step #5: Sniff around example B1 (3/4)

Examining code from examples is one of the best ways to learn

- Example: the `DetectorConstruction()` class

geant4-11.1.2/examples/basic/B1/include/DetectorConstruction.hh

```
class DetectorConstruction : public G4VUserDetectorConstruction
{
  public:
    DetectorConstruction();
    ~DetectorConstruction() override;

    G4VPhysicalVolume* Construct() override;

  // some code
};
```

geant4-11.1.2/examples/basic/B1/src/DetectorConstruction.cc

```
G4VPhysicalVolume* DetectorConstruction::Construct()
{
  // some code

  G4Box* solidWorld = // ...

  G4LogicalVolume* logicWorld = // ...

  G4VPhysicalVolume* physWorld = // ...

  // some code
}
```

Step #5: Sniff around example B1 (4/4)

Examining code from examples is one of the best ways to learn

- Example: the `DetectorConstruction()` class

Users are responsible for creating the simulated geometry

`Geant4` provides *virtual* classes to be inherited by user code that *overrides* the virtual methods

Solids, *logical volumes* and *physical volumes* are created in the `::Construct()` method

geant4-11.1.2/examples/basic/B1/include/DetectorConstruction.hh

```
class DetectorConstruction : public G4VUserDetectorConstruction
{
  public:
    DetectorConstruction();
    ~DetectorConstruction() override;

    G4VPhysicalVolume* Construct() override;

  // some code
};
```

geant4-11.1.2/examples/basic/B1/src/DetectorConstruction.cc

```
G4VPhysicalVolume* DetectorConstruction::Construct()
{
  // some code

  G4Box* solidWorld = // ...

  G4LogicalVolume* logicWorld = // ...

  G4VPhysicalVolume* physWorld = // ...

  // some code
}
```

Step #6: **Basic examples** (1/3)

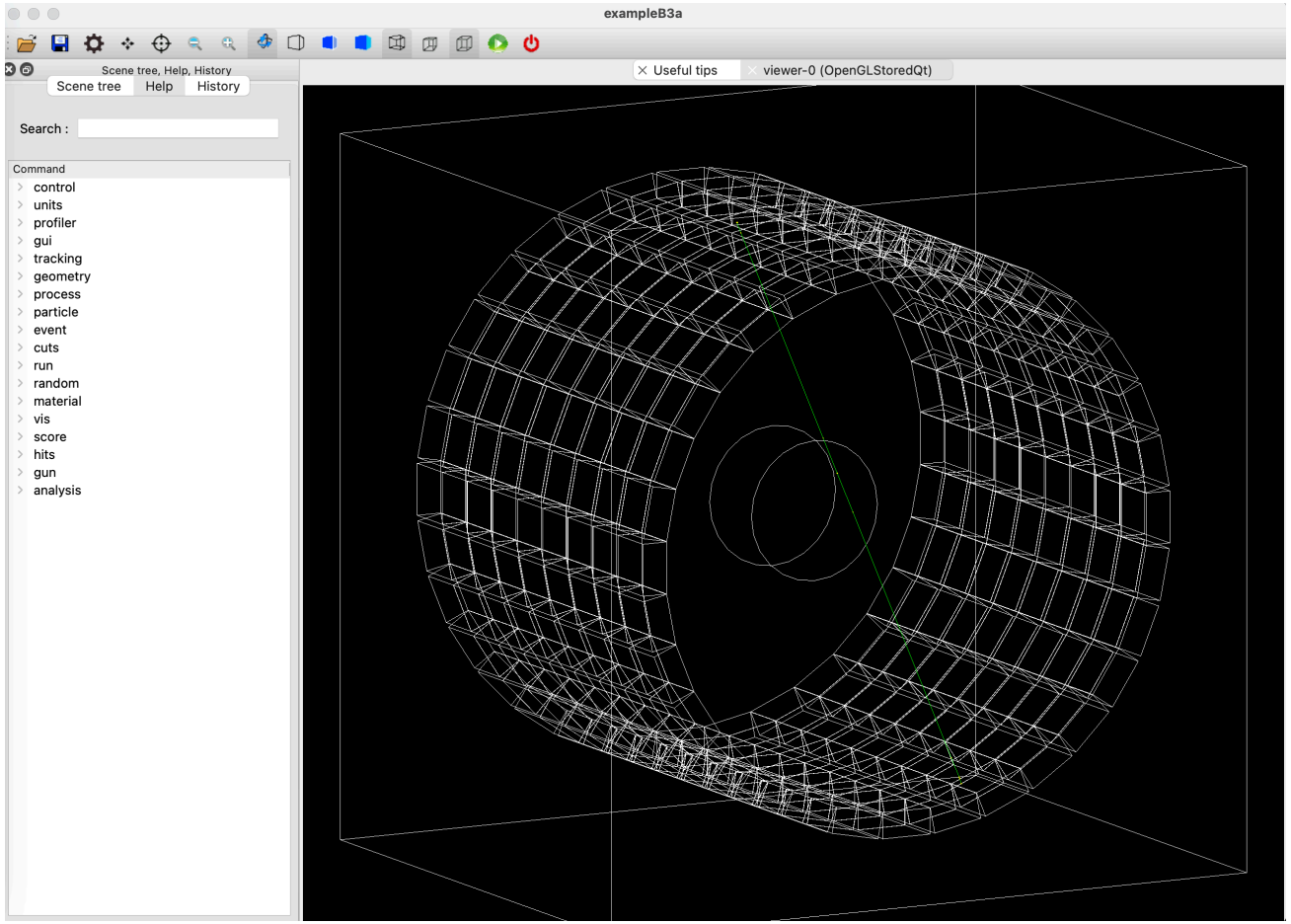
Basic examples demonstrates simple features on simplified geometries (good for learning)

- **Example B1** (previous slides): simple volumes and scoring with *stepping action*
- **Example B2**: magnetic field, scoring with *sensitive detectors* and *hits*, *step limiter*

Step #6: Basic examples (2/3)

Basic examples demonstrates simple features on simplified geometries (good for learning)

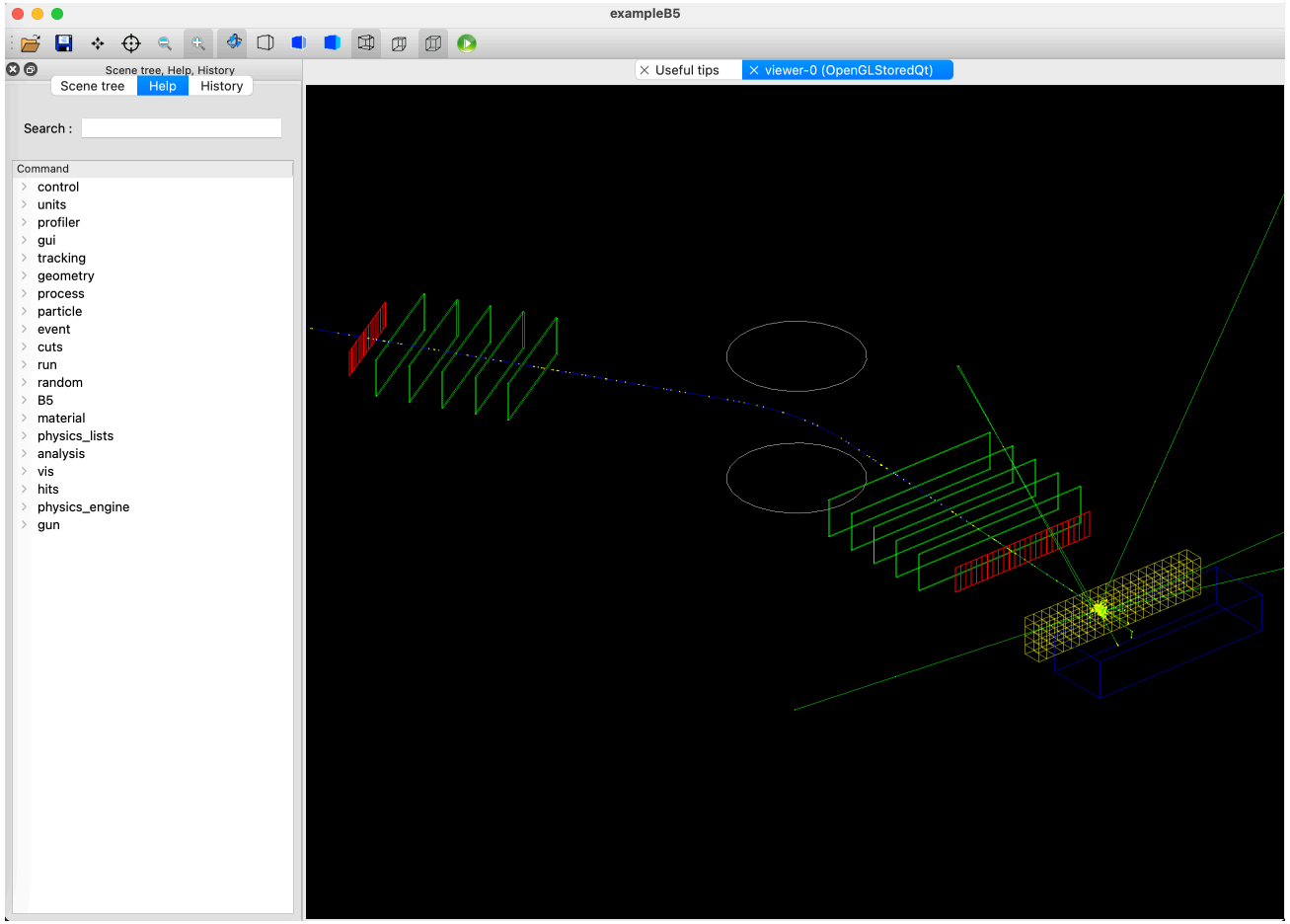
- **Example B1** (previous slides): simple volumes and scoring with *stepping action*
- **Example B2**: magnetic field, scoring with *sensitive detectors* and *hits*, *step limiter*
- **Example B3** (PET system): placement with *rotations*, scoring with *scorers, radioactive source*



Step #6: Basic examples (3/3)

Basic examples demonstrates simple features on simplified geometries (good for learning)

- **Example B1** (previous slides): simple volumes and scoring with *stepping action*
- **Example B2**: magnetic field, scoring with *sensitive detectors* and *hits*, *step limiter*
- **Example B3** (PET system): placement with *rotations*, scoring with *scorers, radioactive source*
- **Example B4**: geometry with *replicas*, saving histograms and ntuples with *g4analysis*
- **Example B5**: *multiple sensitive detectors*, defining *UI commands*, *g4analysis*



equivalent to the Hands On 2~4

Step #7: Extended and advanced examples (1/2)

Extended examples demonstrate specific `Geant4` features and more complex use cases (some requires external libraries)

- They are divided in macro areas: *common, eventgenerator, g3tog4, medical, parameterisations, polarisation, visualization, analysis, electromagnetic, exoticphysics, geometry, optical, persistency, radioactivedecay, biasing, errorpropagation, field, hadronic, parallel, physicslists, runAndEvent*

Step #7: Extended and advanced examples (2/2)

Extended examples demonstrate specific `Geant4` features and more complex use cases (some requires external libraries)

- They are divided in macro areas: *common, eventgenerator, g3tog4, medical, parameterisations, polarisation, visualization, analysis, electromagnetic, exoticphysics, geometry, optical, persistency, radioactivedecay, biasing, errorpropagation, field, hadronic, parallel, physicslists, runAndEvent*

Advanced examples demonstrate complex, real-life solutions from domain-specific communities (HEP, biomedical-physics, space science, …)

- Each example is a standalone `Geant4` simulation targeting a specific application: *doiPET, hadrontherapy, microelectronics, xray_telescope, CaTS, STCyclotron, eRosita, human_phantom, nanobeam, ChargeExchangeMC, air_shower, fastAerosol, iort_therapy, purging_magnet, amsEcal, gammaknife, lAr_calorimeter, radioprotection, HGCal_testbeam, brachytherapy, gammaray_telescope, medical_linac, underground_physics, ICRP110_HumanPhantoms, composite_calorimeter, gorad, microbeam, xray_fluorescence*

Step #8: Refer to the Documentation

The following documents are accessible through the Geant4 website

- **Book for Application Developers** [link]: introduces the first-time user to Geant4, provides a description of the available tools and supply the practical information required to develop and run simulation applications
- **Physics Reference Manual** [link]: presents the theoretical formulation, model, or parameterization of the physics interactions and describes the probability of the occurrence of an interaction and the sampling mechanisms required to simulate it
- **Users Guide for Toolkit Developers** [link]: provides information for those who want to understand or refer to the detailed design of the toolkit, as well as procedures for extending the functionality of the toolkit

Step #9: Refer to the users support channels

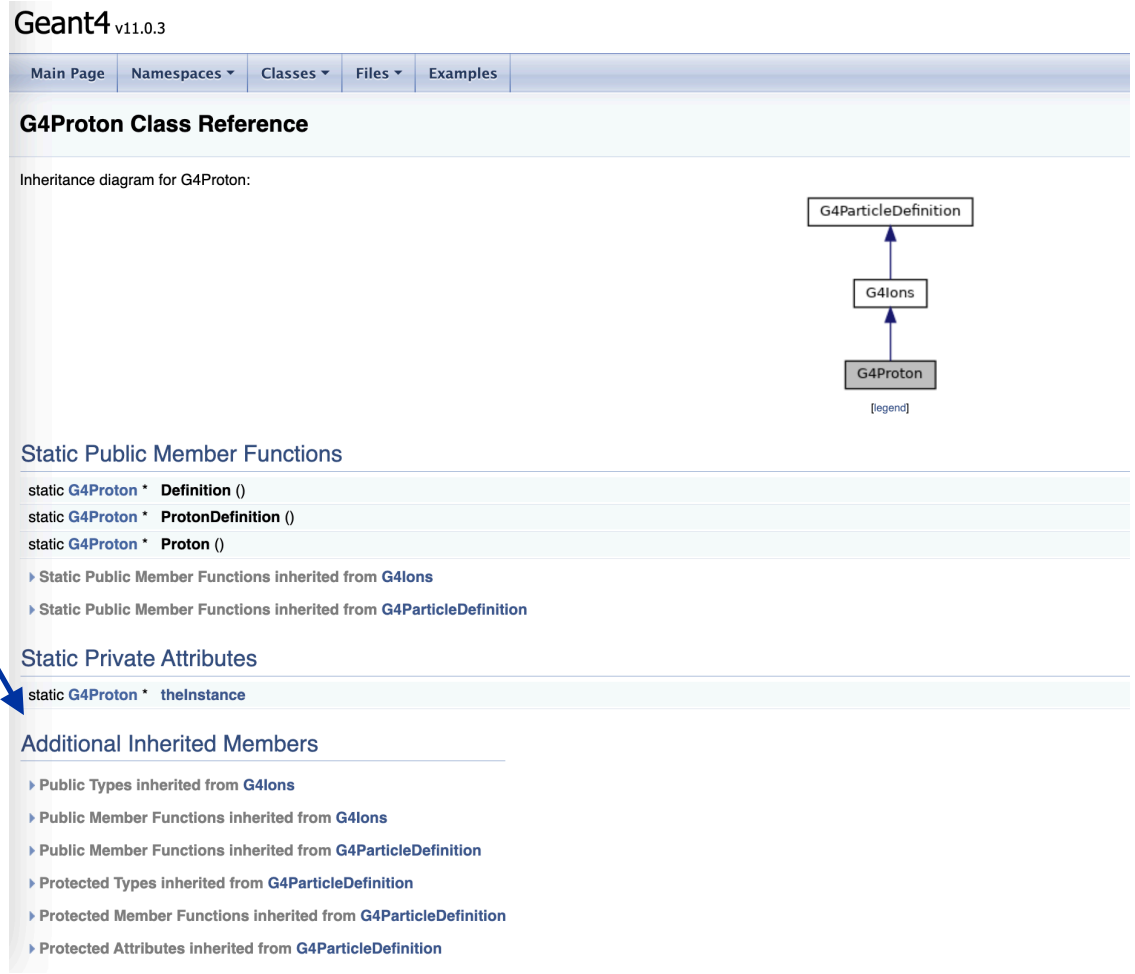
`Geant4` code can be inspected on

- **Doxygen** [link]: every class and file is available and fully *hyper-linked*
  It is useful to overview a class including its *inheritance*

Similarly `Geant4` code is accessible through the LXR Code Browser

- **LXR** [link]

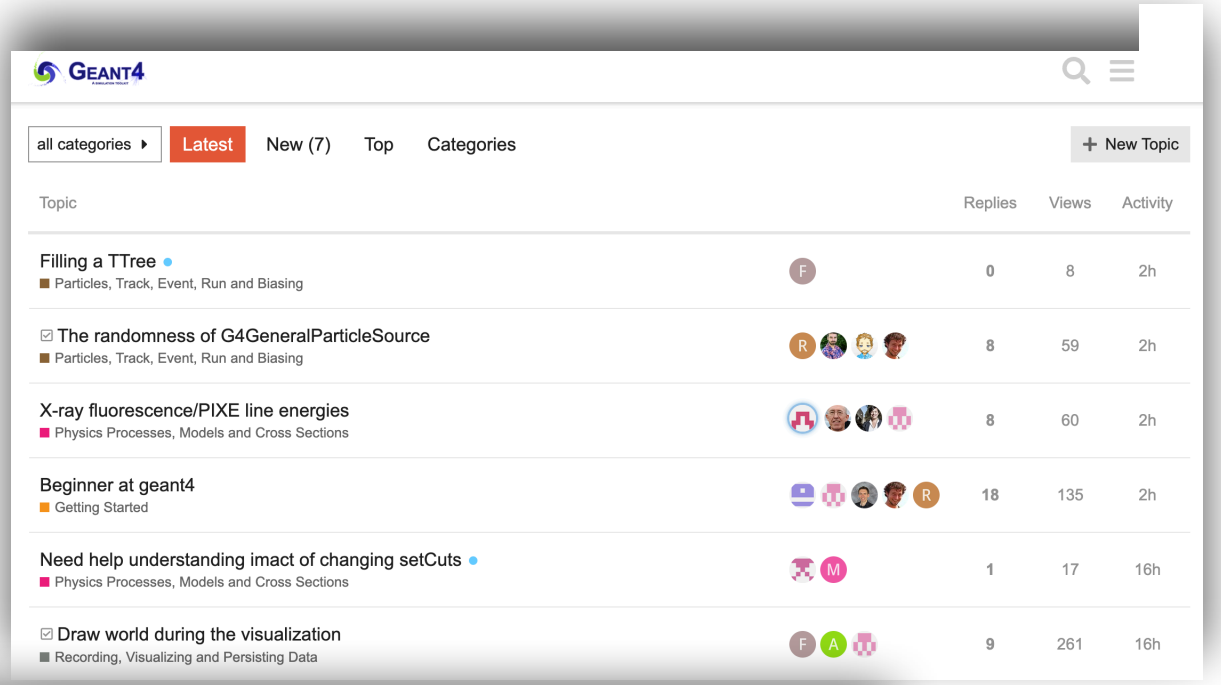
Example: inspect G4Proton class through doxygen

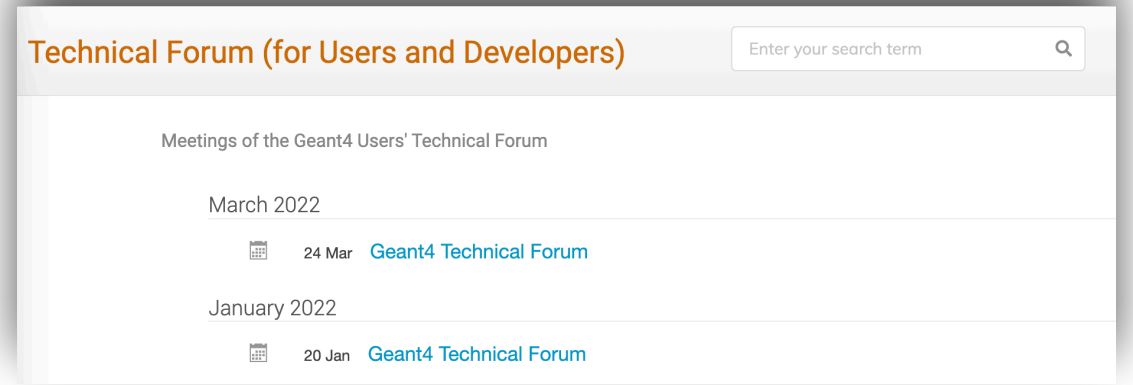


Step #10: Refer to the forums

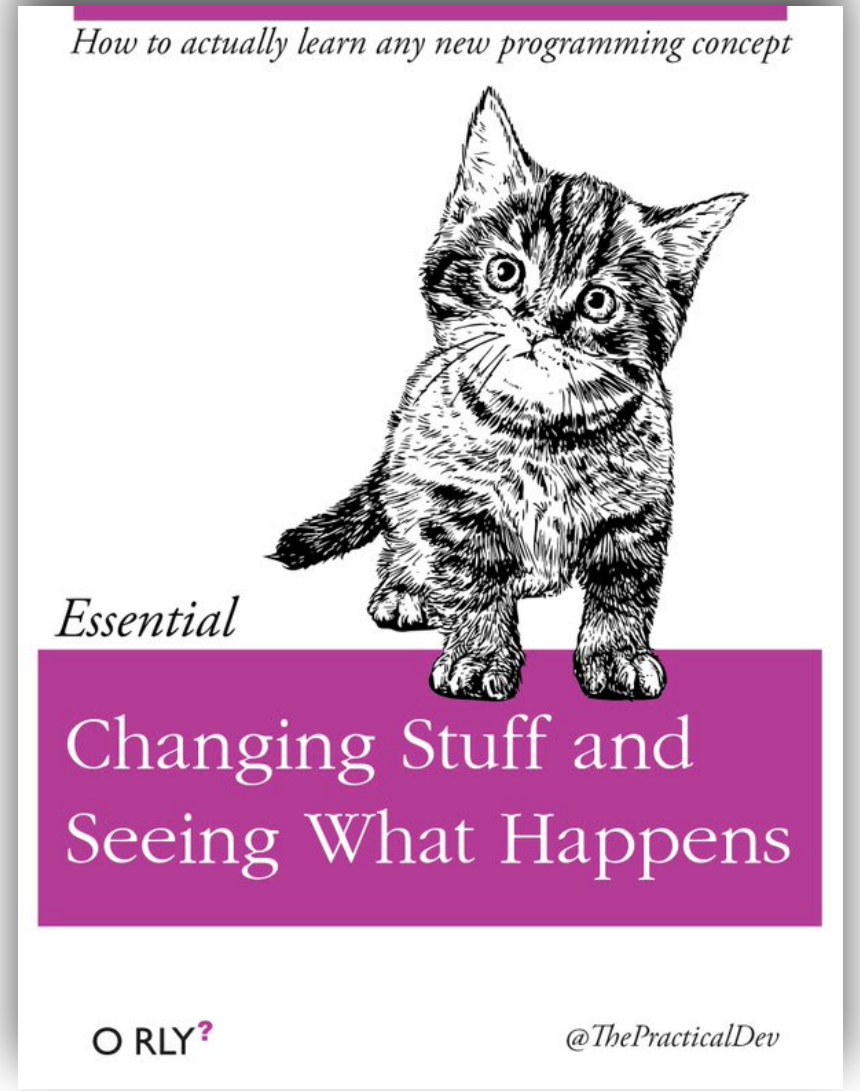
Users can reach the Geant4 *community* via the

- **Geant4 Forum** [link]
  - Topics are divided into 9 categories: *forum issues, getting started, geometry fields transportation, physics list, physics processes models and cross sections, particles track event run biasing, recording visualizing and persisting data, ideas and requirements, applications*
  - Signing-up is required to post a topic
- **Geant4 Technical Forum** [indico]
  - Regular meetings between the Geant4 developers and the users community

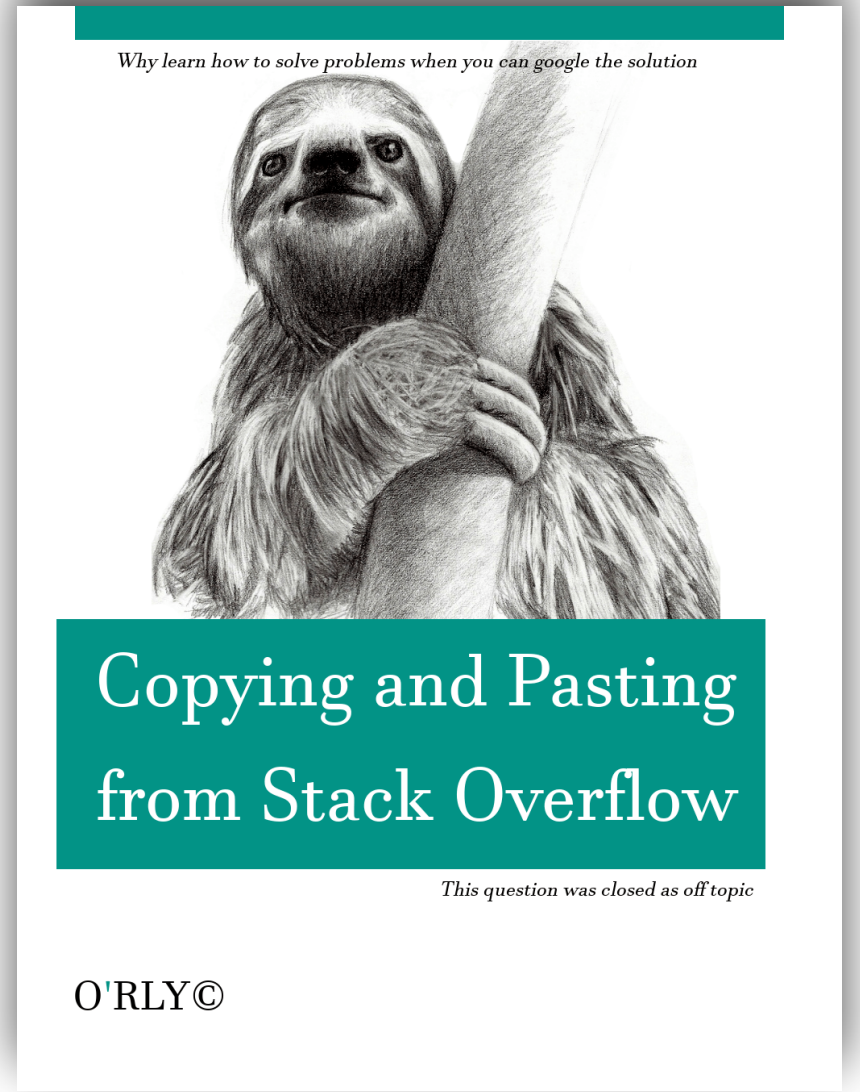




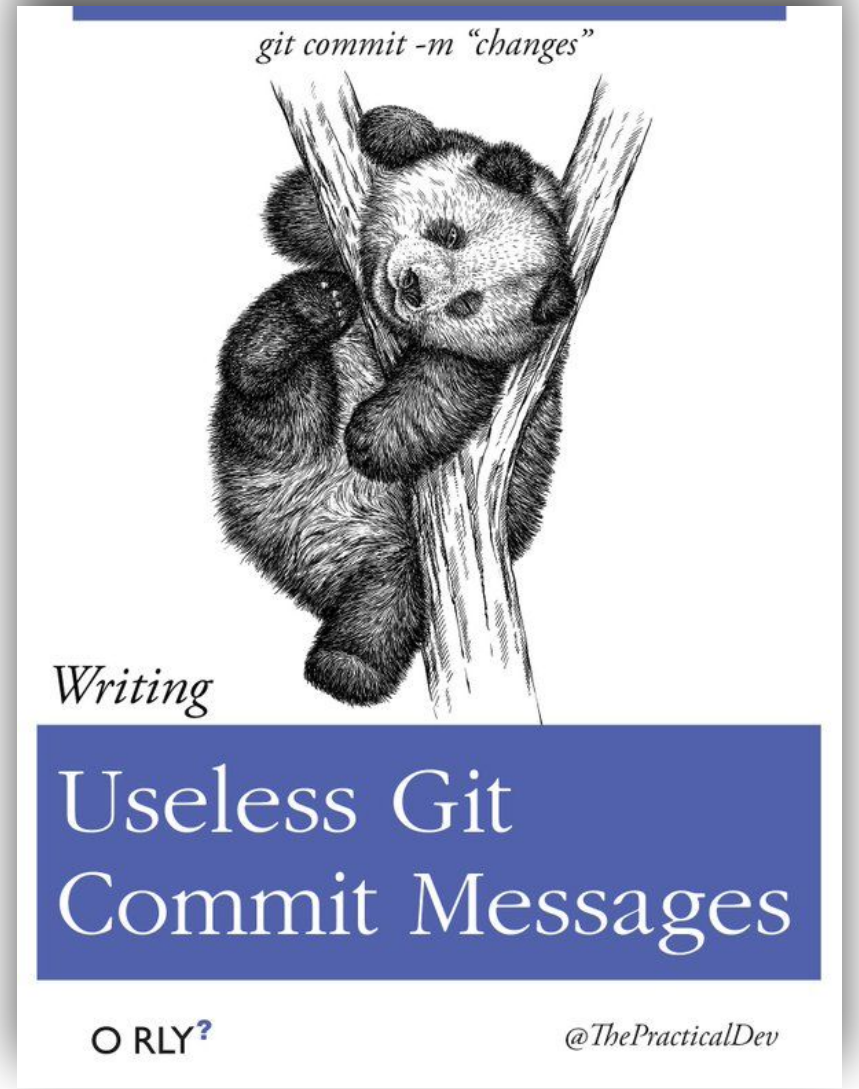
Some recommendations: *how to learn*



Some recommendations: *how to do*



Some recommendations: *try to avoid*



Recap of Examples and Documentation

`Geant4` is a modular code made of ~2 million lines of code → *be patient, it takes time to master it*

Few recommended steps:

- Start from basic examples
  - Isolate their building blocks
  - Adopt the *"change something and see what happens"* mind
- Inspect `Geant4` *classes* on Doxygen
- Refer to the documentation or post your questions on the user forum