연세대학교
YONSEI UNIVERSITY

GEANT4
A SIMULATION TOOLKIT

11th International Geant4 Course in Korea 2024

First Step in Geant4: Examples and Documentation

(based on previous Geant4 courses)

Yeon Soo Yeom, Ph.D.

Department of Radiation Convergence Engineering,

Yonsei University – Mirae Campus

2024.08.20

iDOL
Yonsei Innovative DOsimetry Lab.

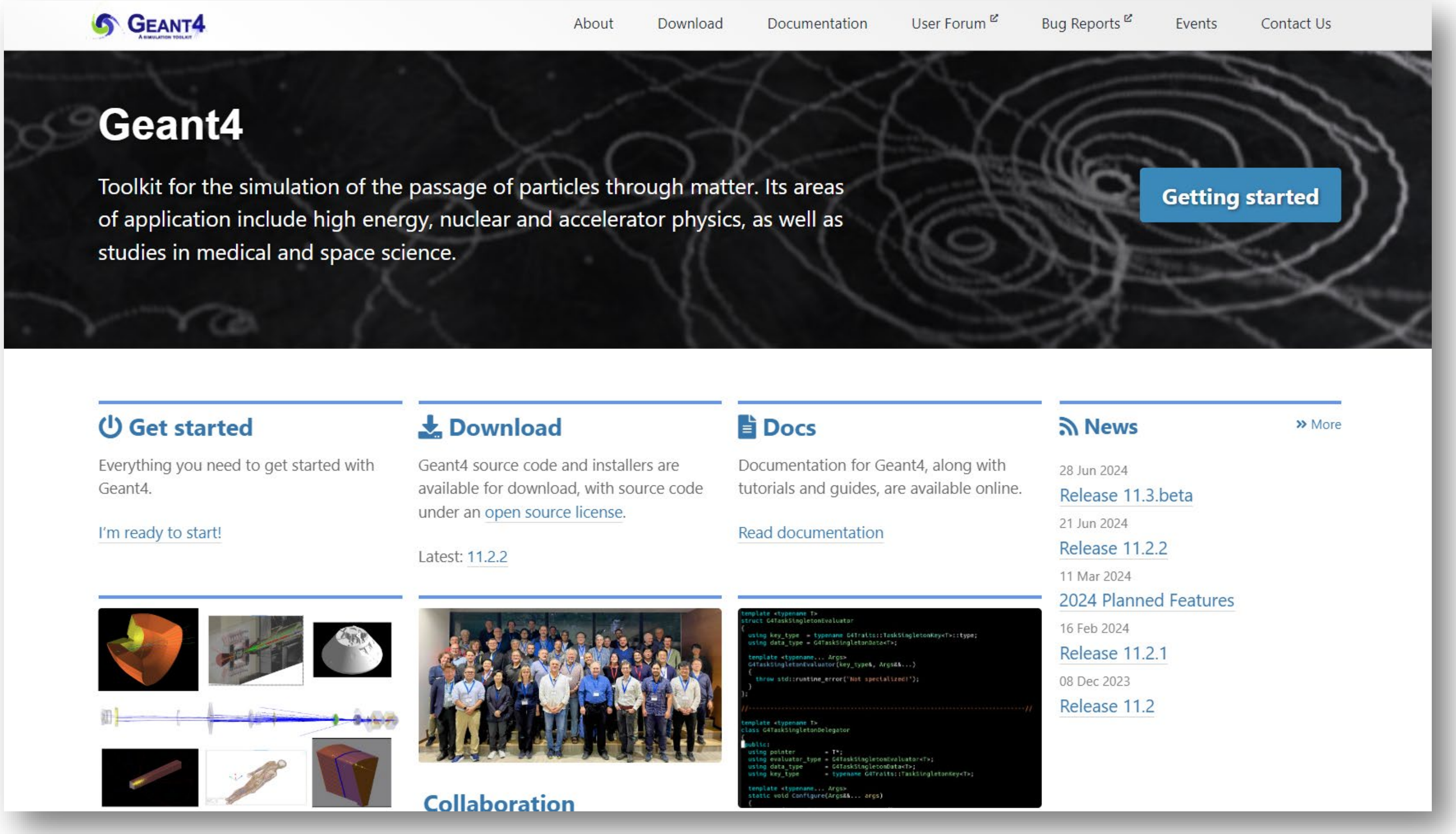
Covered topics

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, …



Geant4

Toolkit for the simulation of the passage of particles through matter. Its areas of application include high energy, nuclear and accelerator physics, as well as studies in medical and space science.

Get started

Everything you need to get started with Geant4.

I'm ready to start!

Download

Geant4 source code and installers are available for download, with source code under an open source license.

Latest: 11.2.2

Docs

Documentation for Geant4, along with tutorials and guides, are available online.

Read documentation

News

» More

28 Jun 2024
Release 11.3.beta

21 Jun 2024
Release 11.2.2

11 Mar 2024
2024 Planned Features

16 Feb 2024
Release 11.2.1

08 Dec 2023
Release 11.2

Collaboration

```
template <typename T>
struct G4TaskSingletonEvaluator
{
  using key_type  = typename G4Traits::TaskSingletonKey<T>::type;
  using data_type = G4TaskSingletonData<T>;

  template <typename... Args>
  G4TaskSingletonEvaluator(key_type&, Args&&...)
  {
    throw std::runtime_error("Not specialized!");
  }
};

//--------------------------------------------------------------------------//

template <typename T>
class G4TaskSingletonDelegator
{
public:
  using pointer        = T*;
  using evaluator_type = G4TaskSingletonEvaluator<T>;
  using data_type      = G4TaskSingletonData<T>;
  using key_type       = typename G4Traits::TaskSingletonKey<T>;

  template <typename... Args>
  static void Configure(Args&&... args)
  {
```

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 geant4-11.2.2.zip

$ mkdir geant4-11.2.2-build; cd geant4-11.2.2-build

$ cmake -DCMAKE_INSTALL_PREFIX=/path-to-install/geant4-11.2.2-install -DGEANT4_INSTALL_DATA=ON -DGEANT4_USE_QT=ON -DGEANT4_BUILD_MULTITHREADED=ON /path-to/geant4-11.2.2/

$ make –j 6

$ 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/geant4-11.2.2-install/bin/geant4.sh

$ cmake -DGeant4_DIR=/path-to/geant4-11.2.2-install/lib/Geant4-11.2.2/
/path-to/geant4-11.2.2/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 file (macro card setting some parameters)

Step #3: Build examples (2/2)

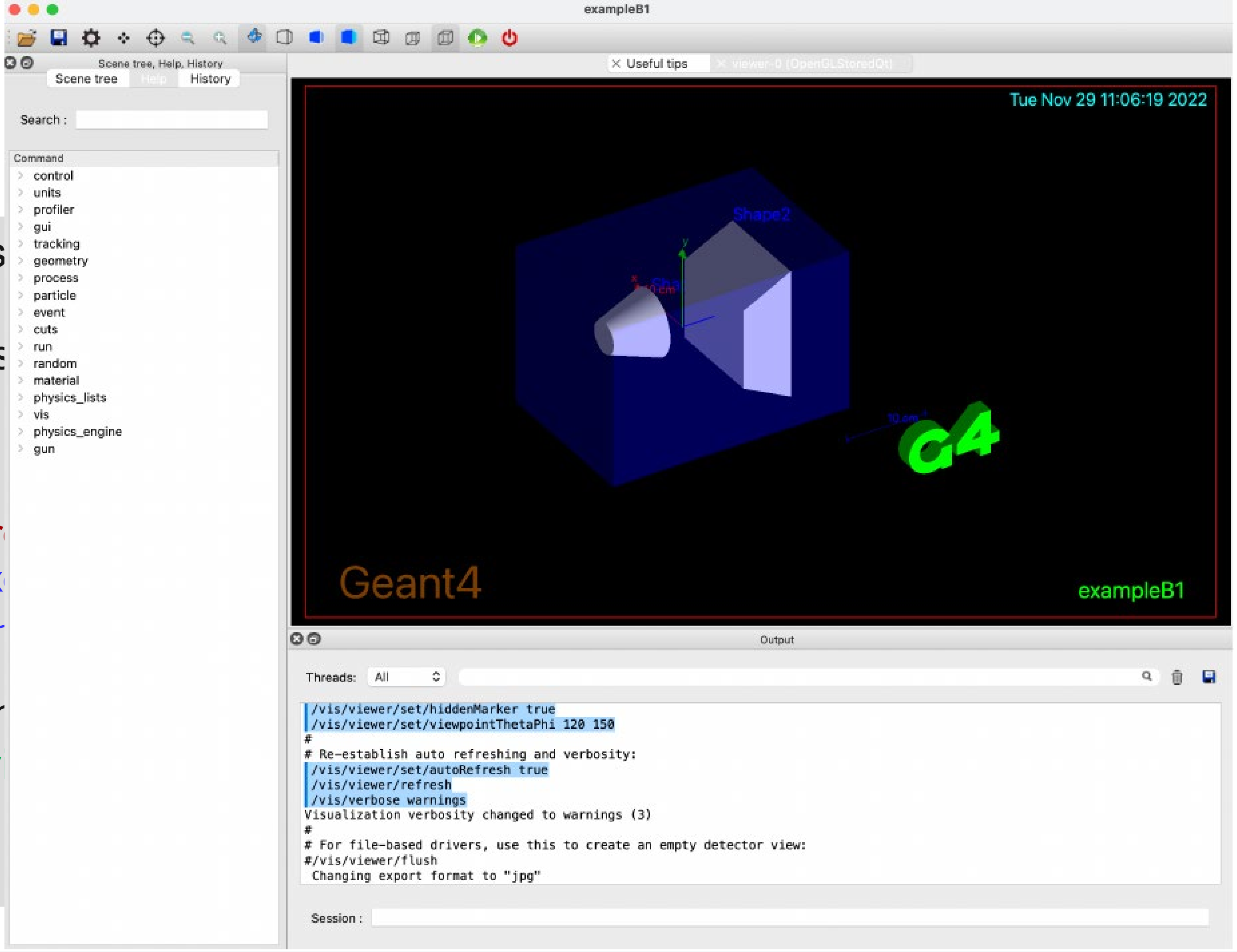
Examples (geant4/examples/) are useful applications to learn Geant4 features (from basic to advanced)

- ❖ Building example B1, the first basic example

Bash – Building example B1

```
$ source /path-to/geant4-11.2.2-install/bin/geant4.s

$ cmake -DGeant4_DIR=/path-to/geant4-11.2.2-ins
/path-to/geant4-11.2.2/examples/basic/B1/

$ make

$ ls

CMakeCache.txt    Makefile    exampleB1    exam
cmake_install.cmake       exampleB1.in       init_v

$ ./exampleB1
```

Several files ar
Look for the ex



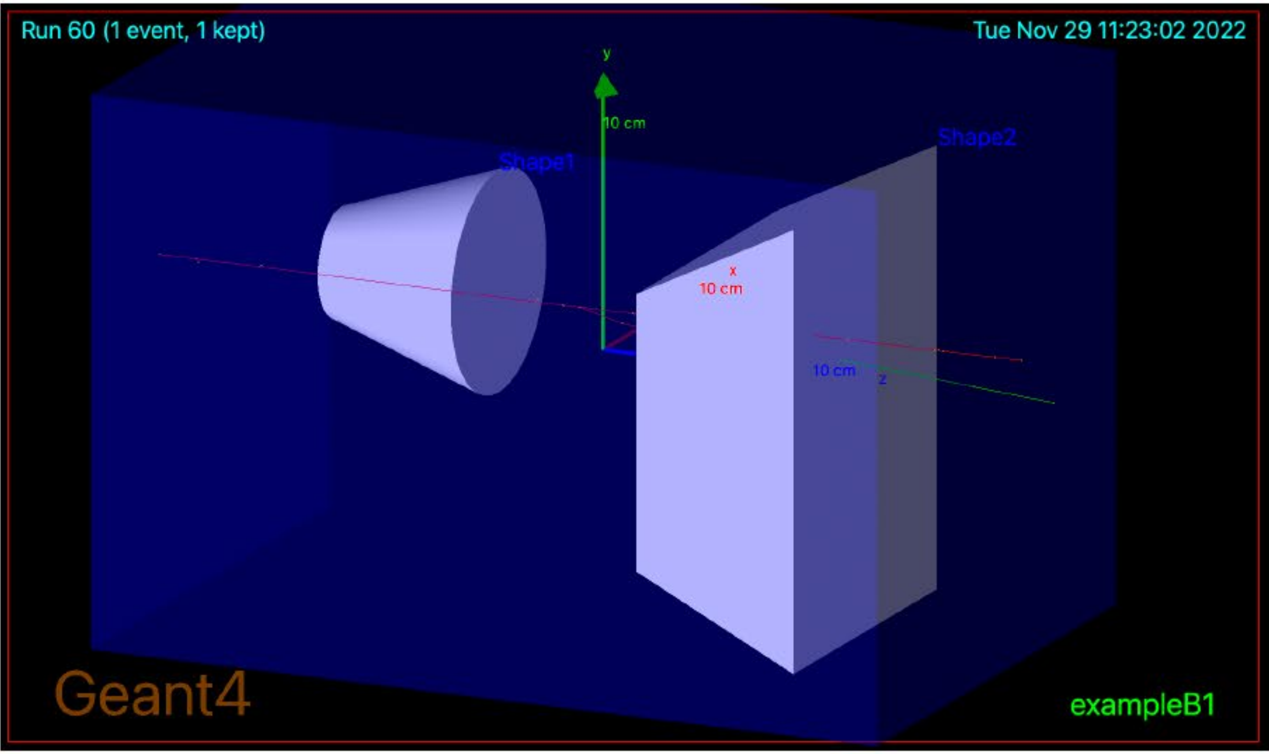
```
/vis/viewer/set/hiddenMarker true
/vis/viewer/set/viewpointThetaPhi 120 150
#
# Re-establish auto refreshing and verbosity:
/vis/viewer/set/autoRefresh true
/vis/viewer/refresh
/vis/verbose warnings
Visualization verbosity changed to warnings (3)
#
# For file-based drivers, use this to create an empty detector view:
#/vis/viewer/flush
 Changing export format to "jpg"
```

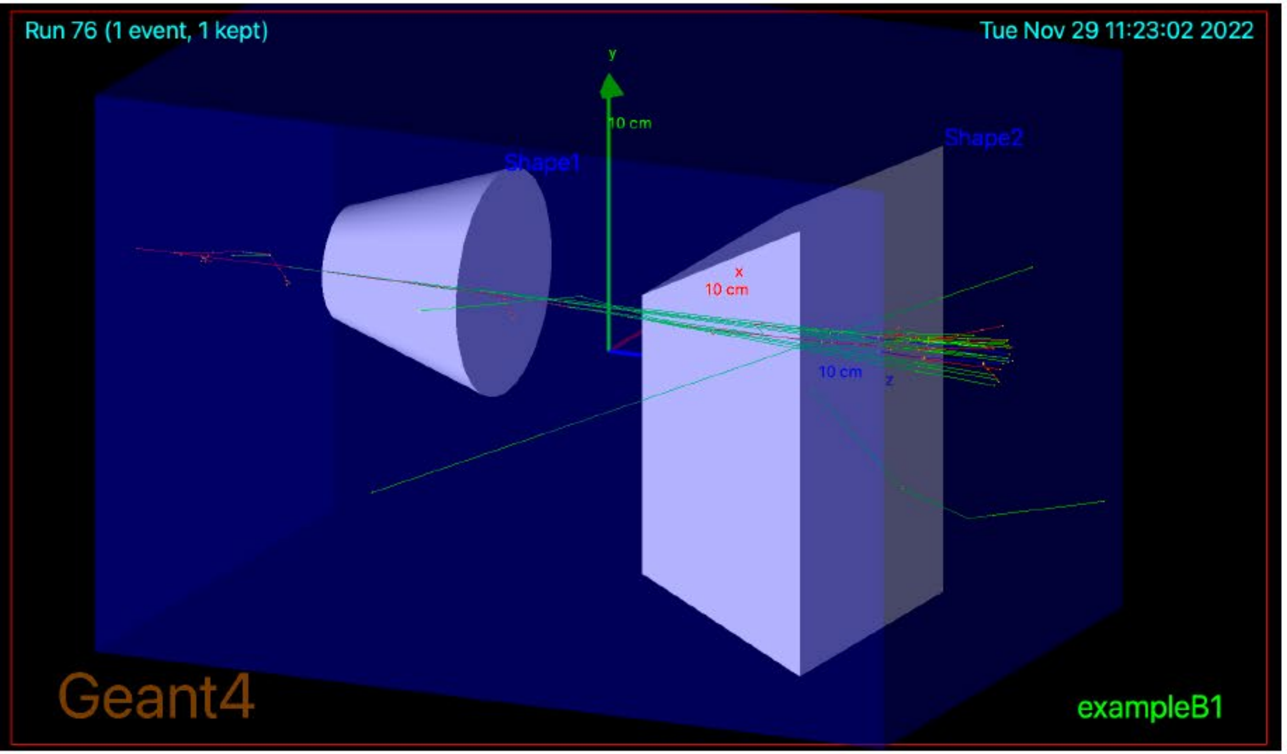
Step #4: Shoot particles

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

geant4-11.2.2/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

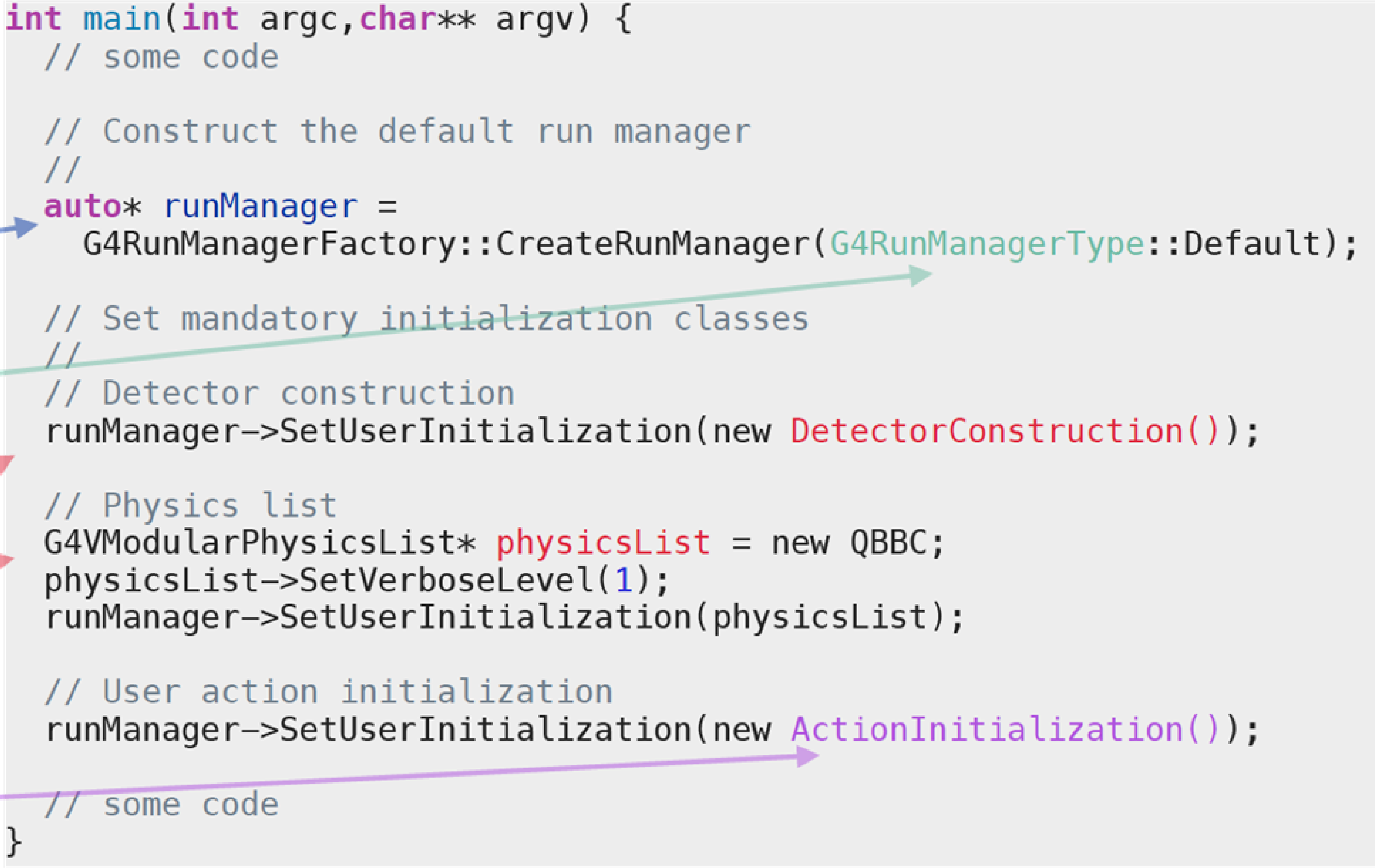
Every Geant4 application has a *run manager*

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…

geant4-11.2.2/examples/basic/B1/exampleB1.cc



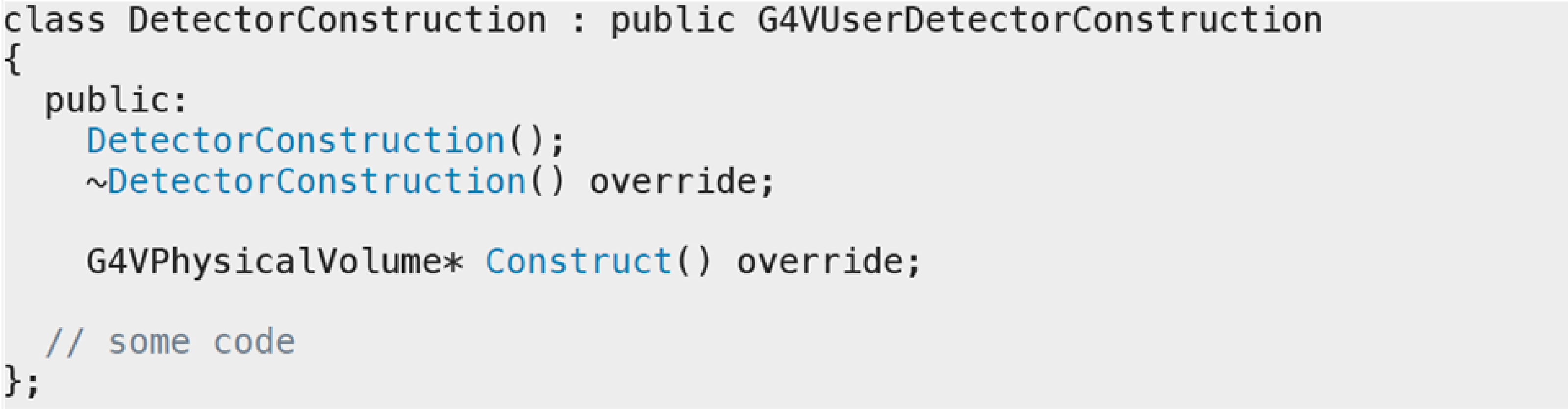
```
int main(int argc,char** argv) {
  // some code

  // 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

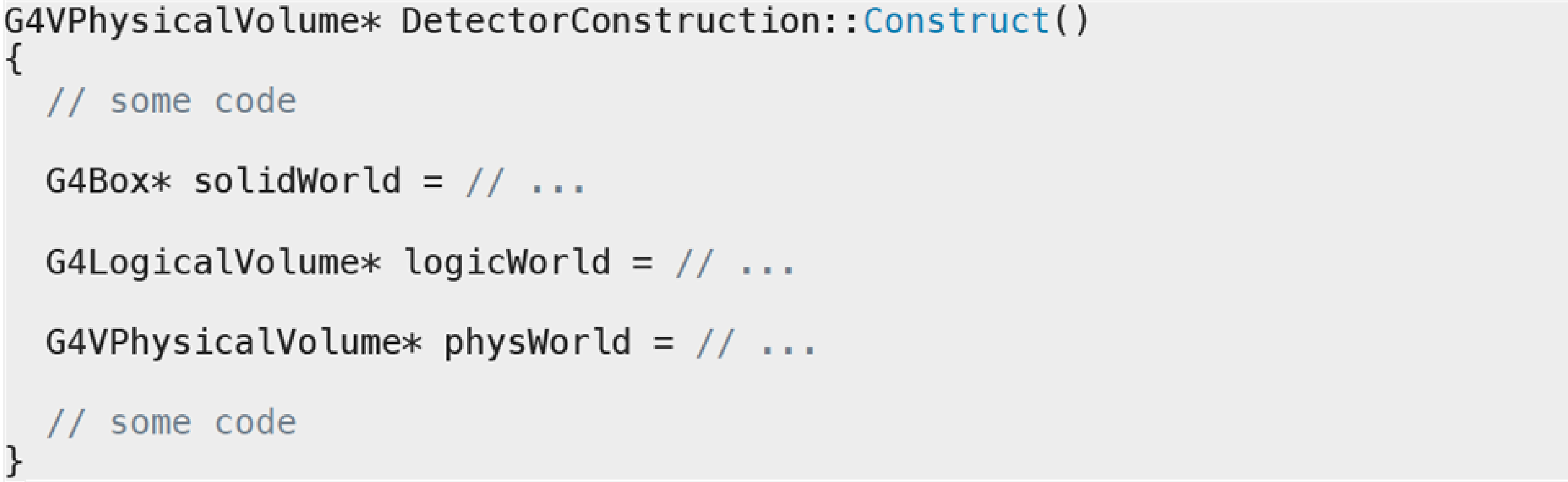
- Example: the `DetectorConstruction()` class

geant4-11.2.2/examples/basic/B1/include/DetectorConstruction.hh



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

    G4VPhysicalVolume* Construct() override;

  // some code
};
```

geant4-11.2.2/examples/basic/B1/src/DetectorConstruction.hh



```
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

geant4-11.2.2/examples/basic/B1/include/DetectorConstruction.hh

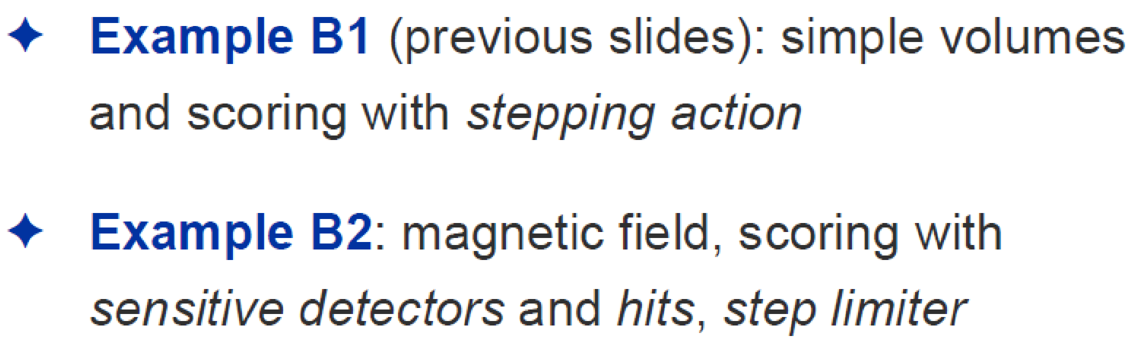
```
class DetectorConstruction : public G4VUserDetectorConstruction
{
  public:
    DetectorConstruction();
    ~DetectorConstruction() override;

    G4VPhysicalVolume* Construct() override;

  // some code
};
```

geant4-11.2.2/examples/basic/B1/src/DetectorConstruction.hh

```
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)

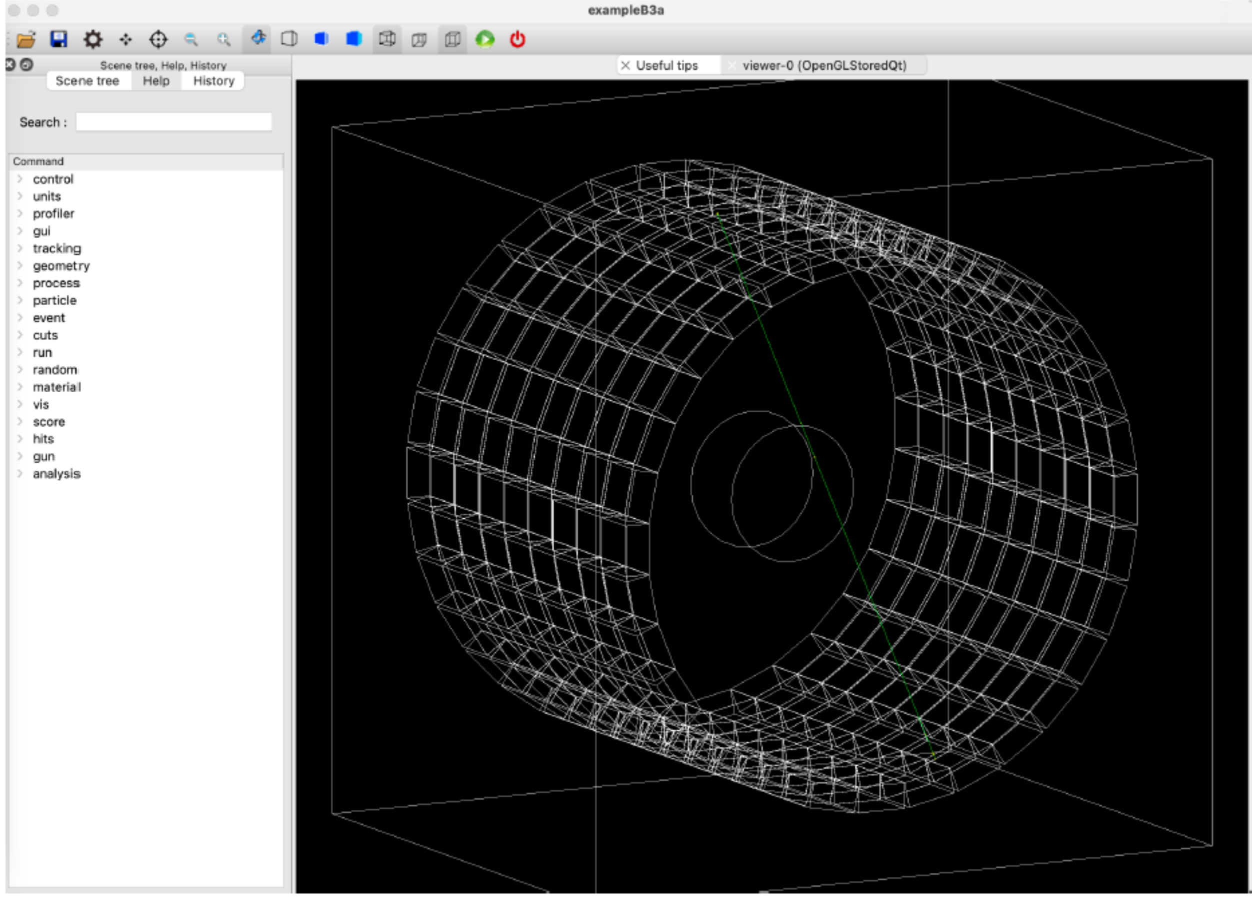


- **Example B1** (previous slides): simple volumes and scoring with *stepping action*
- **Example B2**: magnetic field, scoring with *sensitive detectors* and *hits*, *step limiter*

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*

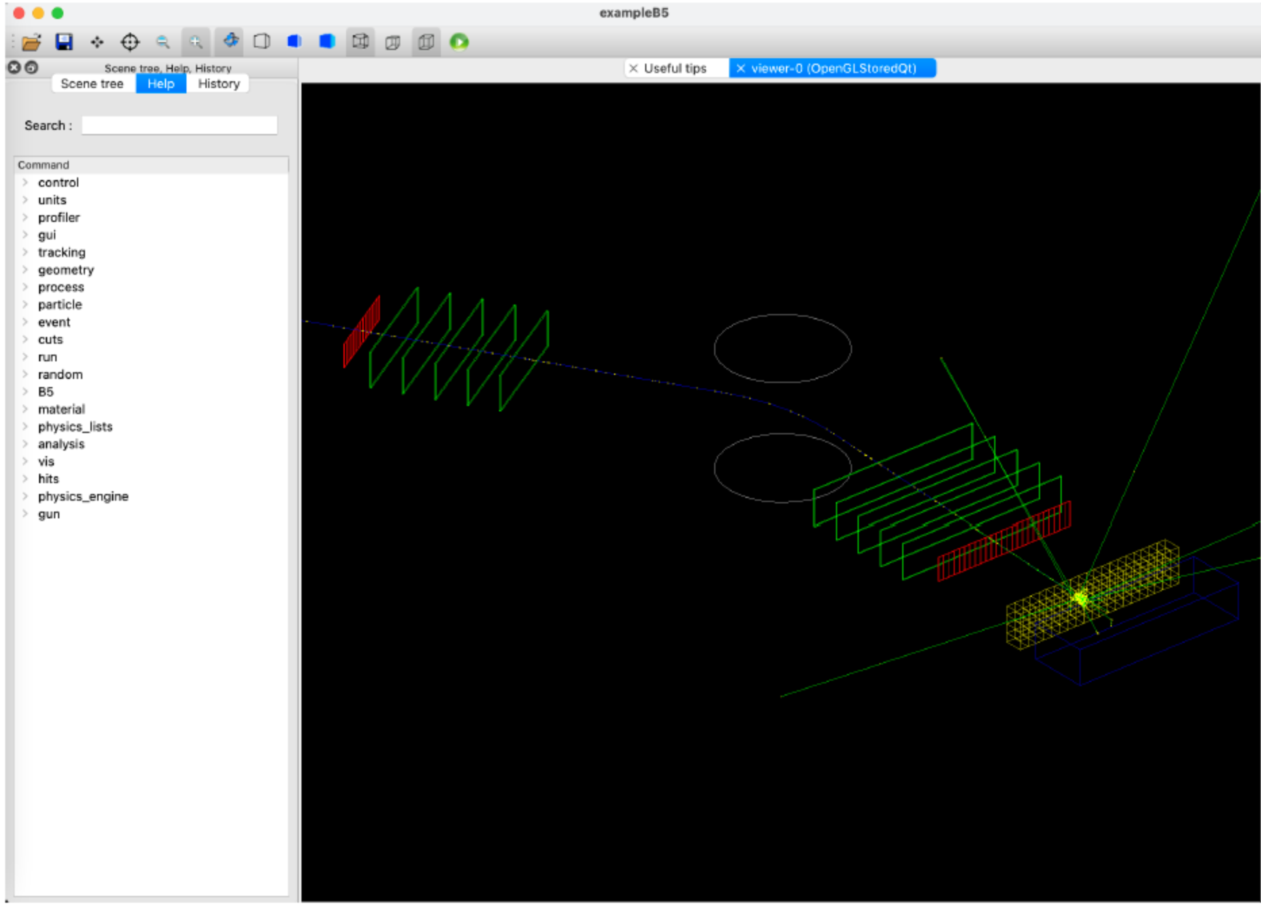


Step #6: Basic examples (2/3)

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*
- **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 (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*

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

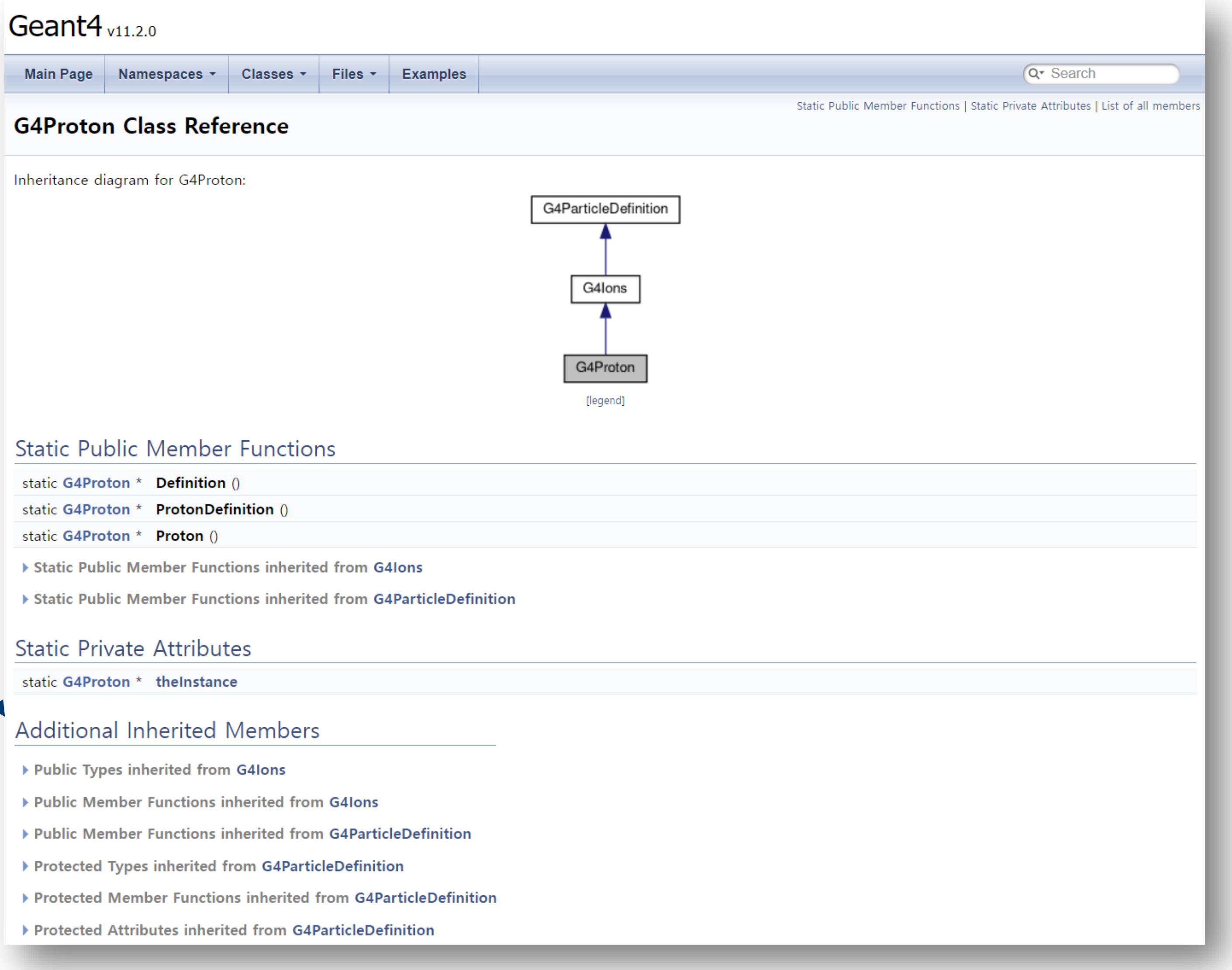
Step #9: Refer to the users support channels

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]



Geant4 v11.2.0

Main Page
Namespaces
Classes
Files
Examples
Search
Static Public Member Functions | Static Private Attributes | List of all members

G4Proton Class Reference

Inheritance diagram for G4Proton:

Static Public Member Functions

static G4Proton *	**Definition** ()
static G4Proton *	**ProtonDefinition** ()
static G4Proton *	**Proton** ()

- Static Public Member Functions inherited from G4Ions
- Static Public Member Functions inherited from G4ParticleDefinition

Static Private Attributes

static G4Proton *	theInstance

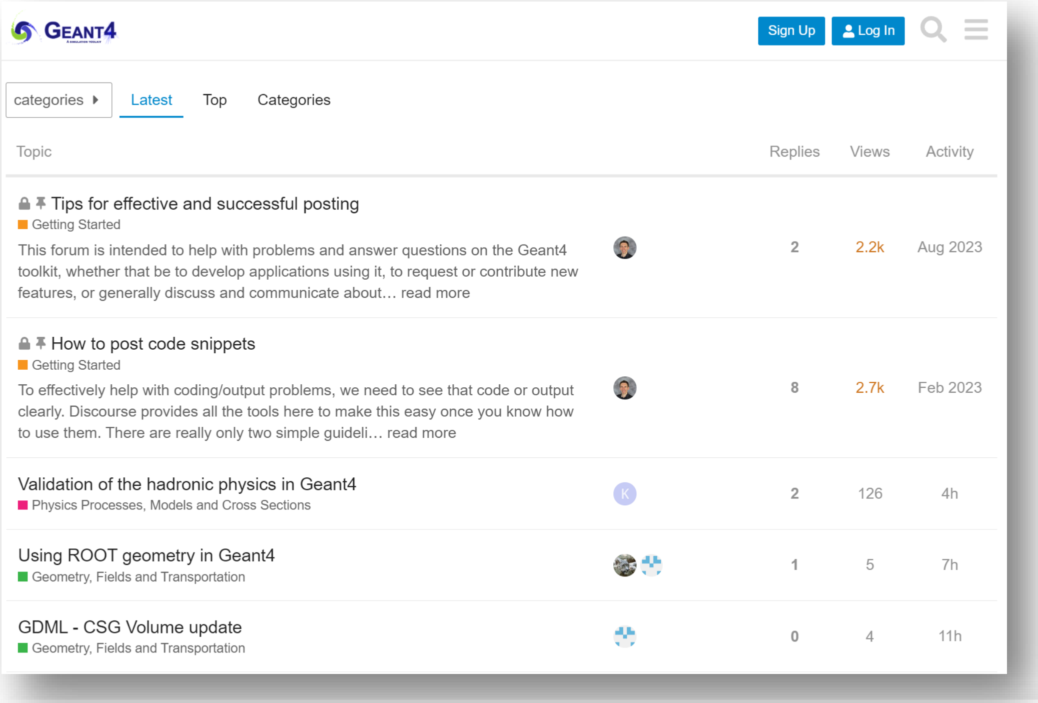
Additional Inherited Members

- Public Types inherited from G4Ions
- Public Member Functions inherited from G4Ions
- Public Member Functions inherited from G4ParticleDefinition
- Protected Types inherited from G4ParticleDefinition
- Protected Member Functions inherited from G4ParticleDefinition
- Protected Attributes inherited from G4ParticleDefinition

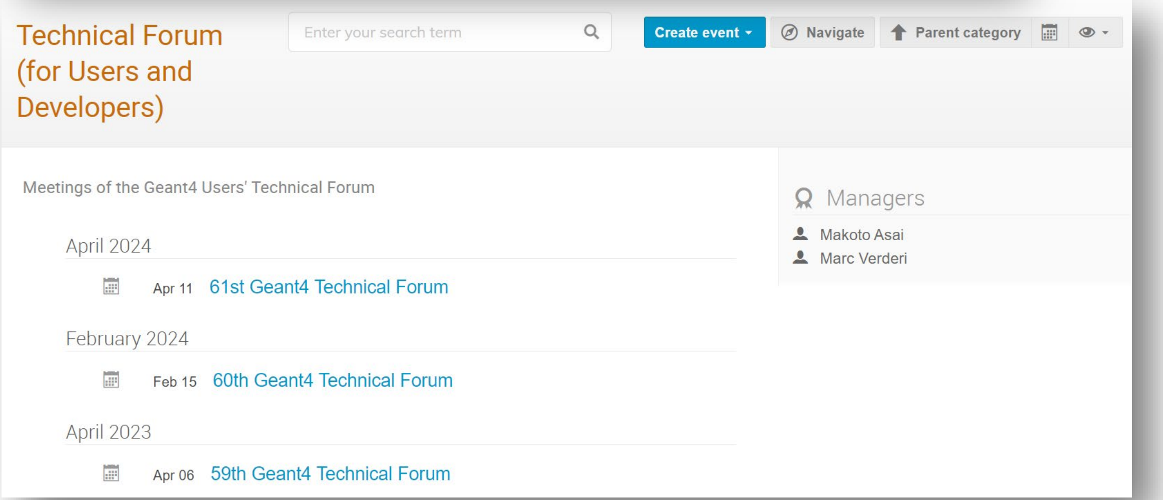
Step #10: Refer to the forums

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



Topic		Replies	Views	Activity
Tips for effective and successful posting Getting Started This forum is intended to help with problems and answer questions on the Geant4 toolkit, whether that be to develop applications using it, to request or contribute new features, or generally discuss and communicate about… read more		2	2.2k	Aug 2023
How to post code snippets Getting Started To effectively help with coding/output problems, we need to see that code or output clearly. Discourse provides all the tools here to make this easy once you know how to use them. There are really only two simple guideli… read more		8	2.7k	Feb 2023
Validation of the hadronic physics in Geant4 Physics Processes, Models and Cross Sections	K	2	126	4h
Using ROOT geometry in Geant4 Geometry, Fields and Transportation		1	5	7h
GDML - CSG Volume update Geometry, Fields and Transportation		0	4	11h



Technical Forum (for Users and Developers)

Meetings of the Geant4 Users' Technical Forum

April 2024

Apr 11 61st Geant4 Technical Forum

February 2024

Feb 15 60th Geant4 Technical Forum

April 2023

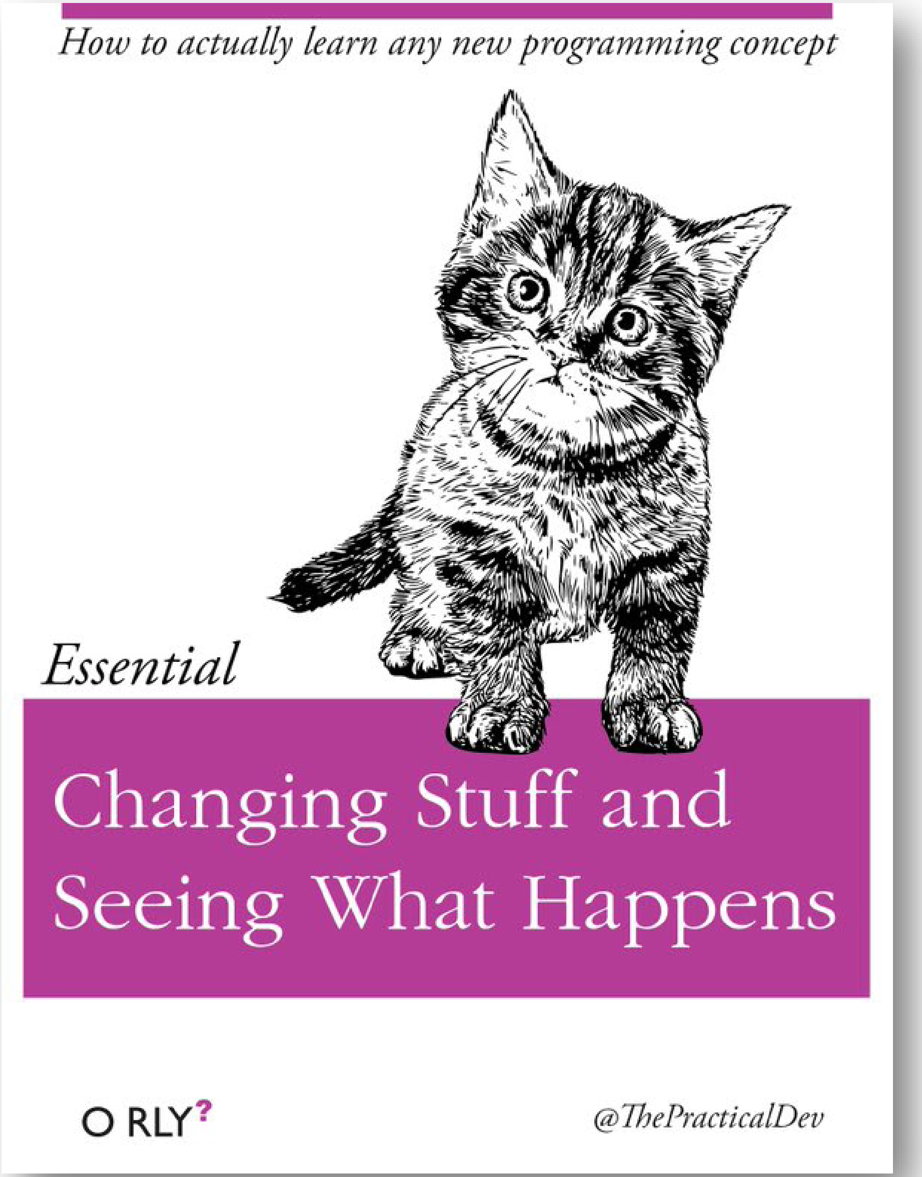
Apr 06 59th Geant4 Technical Forum

Managers

Makoto Asai

Marc Verderi

Some recommendations: *how to learn*



How to actually learn any new programming concept

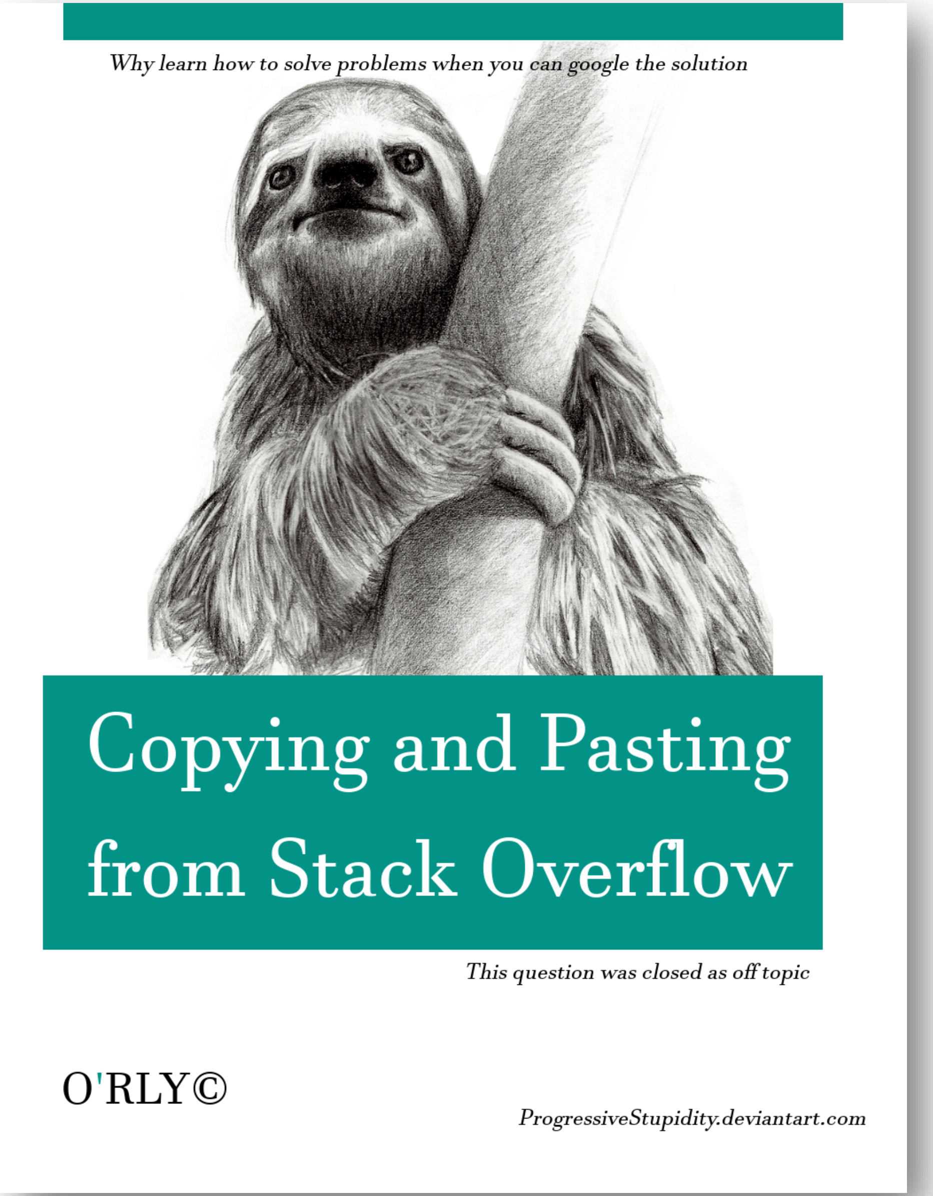
Essential

Changing Stuff and Seeing What Happens

O RLY?

@ThePracticalDev

Some recommendations: *how to do*



Why learn how to solve problems when you can google the solution

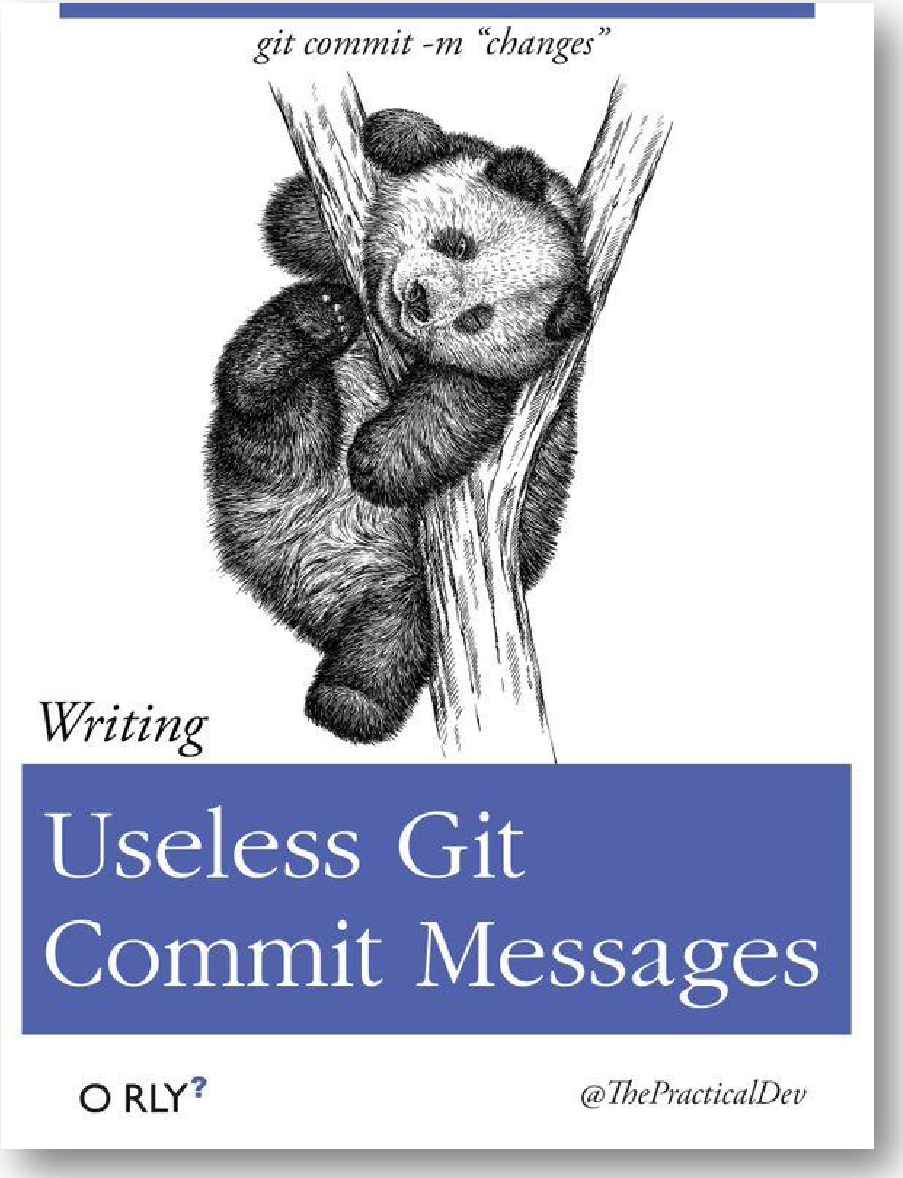
Copying and Pasting from Stack Overflow

This question was closed as off topic

O'RLY©

ProgressiveStupidity.deviantart.com

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

Thanks for your attention