

JUNO GEANT4 SCHOOL

Beijing (北京)
15-19 May 2017

(Graphical) User Interface 1

Geant4 tutorial



3 ways of controlling the simulation

1) **hard-coded** application

- no user interaction
- everything specified in the C++ source
- re-compile needed to apply changes

2) **batch** session

- commands in external macro file

3) **interactive** session

- real-time command input by user
- textual, graphical, (network-based)

Select the way of control

main.cc

```
int main(int argc, char** argv) {
    G4RunManager* runManager = new G4RunManager;
    runManager->SetUserInitialization(new MyDetectorConstruction());

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

    // User actions initialization
    runManager->SetUserInitialization(new MyActionInitialization());
```

Insert the control code here!

```
delete runManager;
}
```

① Hard-coded C++

```
// ...  
// User actions initialization  
runManager->SetUserInitialization(new MyActionInitialization());  
  
runManager->Initialize();  
runManager->BeamOn(!000);  
  
// ...  
delete runManager;  
}
```

- You must initialize and start the run by issuing “beam on”.
- Even the number of events has to be specified!

② Batch session

```
// ...
// User actions initialization
runManager->SetUserInitialization(new MyActionInitialization());

G4UImanager* UImanager = G4UImanager::GetUIpointer();
G4String command = "/control/execute ";
G4String fileName = argv[1];
UImanager->ApplyCommand(command + fileName);

// ...
delete runManager;
}
```

- This example gets the file name of the macro from the command-line argument:

```
./myApplication my-macro.mac
```

Macro file: example basic/B1

exampleB1.mac

```
# Initialize the run (necessary)
```

```
/run/initialize
```

```
# gamma 6 MeV
```

```
/gun/particle gamma
```

```
/gun/energy 6 MeV
```

```
/run/printProgress 100
```

```
/run/beamOn 1000
```

```
# proton 210 MeV
```

```
/gun/particle proton
```

```
/gun/energy 210 MeV
```

```
/run/beamOn 1000
```

This is a comment line
starting with hash, #

Meaning of these
commands will be
explained later 😊

The contents of the file, excluding **#comments**, are executed line by line in the application (from previous slide).

③ Interactive session

- a) Many different session types, inheriting from **G4UIsession** class:
 - command-line based
 - graphical
 - special
 - your own? 😊
- b) **G4UIExecutive** class enabling to select the appropriate session at runtime, based on the environment variables (recommended).

a) Concrete UI session

```
// ...  
  
G4UIsession* session = new G4UITerminal();  
session->StartSession();  
delete session;  
  
// ...
```

Select one of the
concrete classes

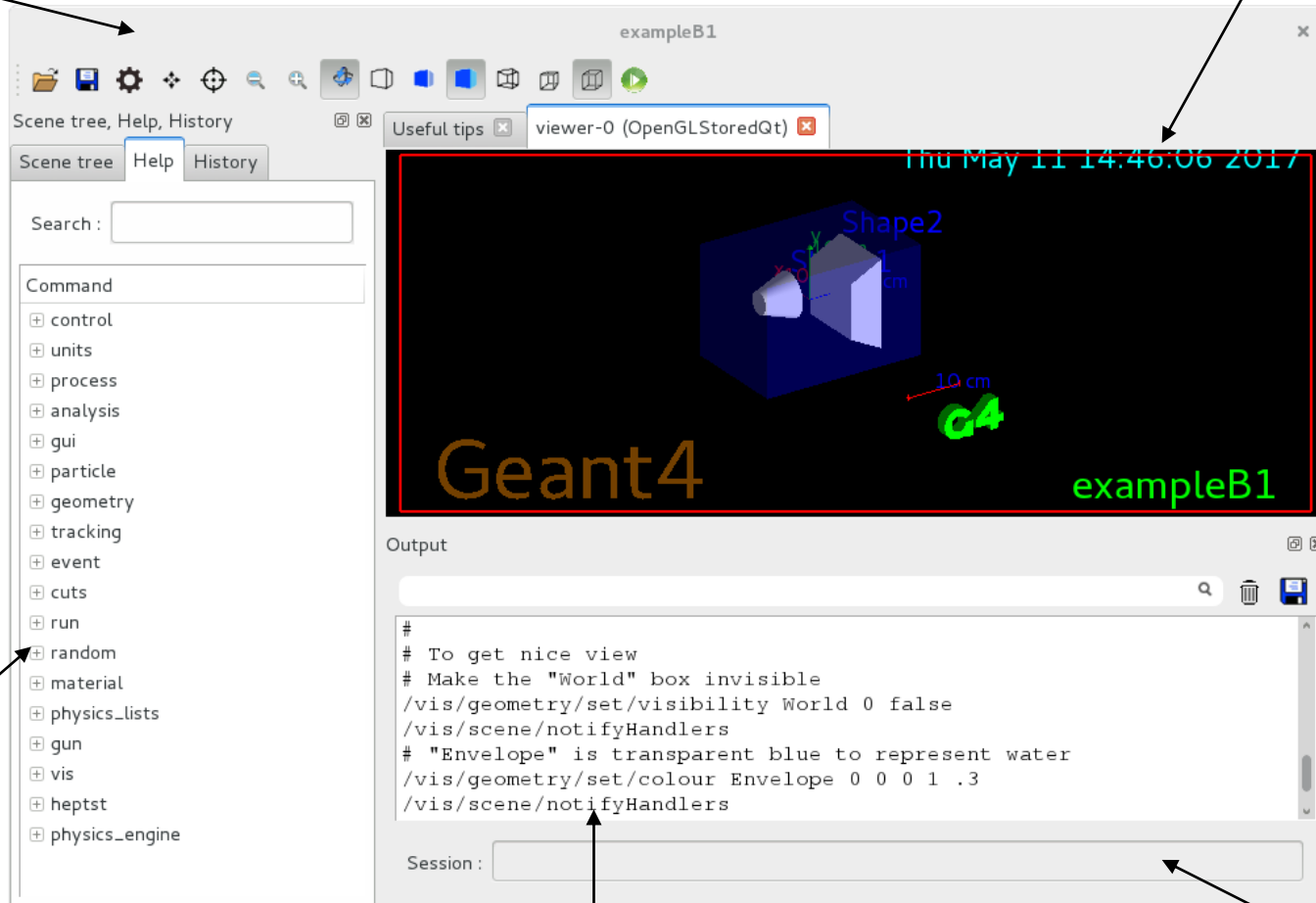
Session types:

- **G4UITerminal** – command-line (like C-shell)
- **G4UI(t)csh** – csh- or tcsh-like specific terminal
- **G4UIXm** – motif-based graphical UI (deprecated)
- **G4UIQt** – modern graphical UI (recommended)
- **G4UIWin32** – for windows only
- **G4UIWt** – experimental web-browser based
- **G4UIGAG** – for GAG java UI

G4UIQt session

Icons

Visualization



Command tree

Output (Cout)

Command input

b) G4UIExecutive

G4UIExecutive *behaves like a* G4UISession, but it selects the most appropriate concrete session:

1. from constructor argument
2. from environment variables: G4UI_USE_QT, ...
3. from \$HOME/.g4Session file
4. from the list (first that applies): Qt, tcsh, Xm

```
// ...  
  
G4UIExecutive* ui = new G4UIExecutive(argc, argv);  
ui->SessionStart();  
delete ui;  
  
// ...
```

You may add a third argument here, i.e. the session name

②③ Universal batch/interactive approach

```
int main(int argc, char** argv) {
    // ...
    if (argc == 1) {
        // Batch mode
        G4UImanager* UImanager = G4UImanager::GetUIpointer();
        G4String command = "/control/execute ";
        G4String fileName = argv[1];
        UImanager->ApplyCommand(command + fileName);
    } else {
        // Interactive mode
        G4UIExecutive* ui = new G4UIExecutive(argc, argv);
        ui->SessionStart();
        delete ui;
    }
    // ...
}
```

- Mode selected based on application argument:
 - no argument = **interactive** mode
 - one argument = **batch** mode

External GUI tools

- HepRApp: Event viewer
 - <http://geant4.slac.stanford.edu/Presentations/vis/G4HepRAppTutorial/G4HepRAppTutorial.html>

Dead tools?

- GGE: Graphical geometry editor
 - <http://erpc1.naruto-u.ac.jp/~geant4>
- GPE: Graphical physics editor
 - <http://erpc1.naruto-u.ac.jp/~geant4>
- OpenScientist: interactive environment for analysis
 - <http://www.lal.in2p3.fr/OpenScientist>

Executing macro commands

1) **hard-coded**

```
// ...  
G4UImanager* UImanager = G4UImanager::GetUIpointer();  
G4String command = "put your command here";  
UImanager->ApplyCommand(command);  
// ...
```

2) **batch** session

- put the command in the macro file

3) **interactive** session

- just type command in the window or terminal

Example UI commands

- `/run/verbose 1` – sets how much output the run manager will print (similar for other classes)
- `/run/initialize` – initializes the run (constructing the geometry, physics and preparing the user actions)
- `/run/beamOn 100` – starts a run with 100 events
- `/control/execute macroName` – run the commands in a macro file
- A complete list of built-in commands is available in the Geant4 Application Developers Guide, Chapter 7.1 (<http://geant4.cern.ch/G4UsersDocuments/UsersGuides/ForApplicationDeveloper/html/Control/commands.html>)

The end

Visualization

- To enable visualization, instantiate a **G4VisExecutive** and invoke its **Initialize()** method
- Geant4 provides interfaces to various graphics (and “graphics”) drivers:
 - OpenGL (+Qt)
 - HepRApp
 - Dawn
 - Wired
 - RayTracer
 - OpenInventor
 - VRML
 -

```
#include <G4VisExecutive.hh>
// ...

int main(int argc, char** argv) {
    // ...
    G4VisManager* visManager = new G4VisExecutive;
    visManager->Initialize();
    G4UIExecutive* ui = new G4UIExecutive(argc, argv);
    ui->SessionStart();
    delete ui;
    delete visManager;
    delete runManager;
}
```