Build an application

Ruhani Khanna (INFN-LNS)

XVI Seminar on Software for Nuclear, Sub-nuclear and Applied Physics
Monday 27th May, 2019 – Alghero, Italy

How to build an application

- Properly organise your code into directories
- Prepare a CMakeLists.txt file
- Create a build directory and run Cmake
- Compile (using the make command) the application
- Run the application

Application source structure in Geant4

Official basic/B1 example:

```
2,4K  4 Dic 14:48 CMakeLists.txt
475B  4 Dic 14:48 GNUmakefile
2,8K  4 Dic 14:48 History
7,5K  4 Dic 14:48 README
4,0K  4 Dic 14:48 exampleB1.cc
226B  4 Dic 14:48 exampleB1.in
 35K  4 Dic 14:48 exampleB1.out
272B  4 Dic 14:49 include
338B  4 Dic 14:48 init_vis.mac
553B  4 Dic 14:48 run1.mac
448B  4 Dic 14:48 run2.mac
272B  4 Dic 14:49 src
3,8K  4 Dic 14:48 vis.mac
```

Macro file containing the commands

The text file CMakeLists.txt is the

Header -> declaration of methods

-> any classes from which it inherits

-> public or not

Source -> code of each method

Involves use of G4RunManager

Header files

```
2,2K  4 Dic 14:48 B1ActionInitialization.hh
2,4K  4 Dic 14:48 B1DetectorConstruction.hh
2,4K  4 Dic 14:48 B1EventAction.hh
2,7K  4 Dic 14:48 B1PrimaryGeneratorAction.hh
2,5K  4 Dic 14:48 B1RunAction.hh
2,4K  4 Dic 14:48 B1SteppingAction.hh
```

Source files

```
2,9K  4 Dic 14:48 B1ActionInitialization.cc
7,7K  4 Dic 14:48 B1DetectorConstruction.cc
2,6K  4 Dic 14:48 B1EventAction.cc
4,3K  4 Dic 14:48 B1PrimaryGeneratorAction.cc
5,8K  4 Dic 14:48 B1RunAction.cc
3,2K  4 Dic 14:48 B1SteppingAction.cc
```

CMake

- **Cmake** is a build configuration tool
- It takes configuration file (CMakeLists.txt)
- It finds all the dependencies (in our case, GEANT4)
- Creates **Makefile** to run the compilation itself

- You have to write this CMakeLists.txt file

Cmakelist.txt

```
cmake_minimum_required(VERSION 2.6 FATAL_ERROR)
project(B1)
option(WITH_GEANT4_UIVIS "Build example with Geant4 UI and Vis drivers" ON)
if(WITH_GEANT4_UIVIS)
  find_package(Geant4 REQUIRED ui_all vis_all)
else()
  find_package(Geant4 REQUIRED)
endif()

include(${Geant4_USE_FILE})
include_directories(${PROJECT_SOURCE_DIR}/include)

file(GLOB sources ${PROJECT_SOURCE_DIR}/src/*.cc)
file(GLOB headers ${PROJECT_SOURCE_DIR}/include/*.hh)

add_executable(exampleB1 exampleB1.cc ${sources} ${headers})
target_link_libraries(exampleB1 ${Geant4_LIBRARIES})

set(EXAMPLEB1_SCRIPTS
  exampleB1.in
  exampleB1.out
  init_vis.mac
  run1.mac
  run2.mac
  vis.mac
  )

foreach(_script ${EXAMPLEB1_SCRIPTS})
  configure_file(
    ${PROJECT_SOURCE_DIR}/${_script}
    ${PROJECT_BINARY_DIR}/${_script}
    COPYONLY
    )
```

File structure

Cmake minimum version and Project name

Find and configure G4

Configure the project to use G4 and B1 headers

List the sources

Define and link the executable

Copy any macro files to the build directory

Include -> configures header path and compiler flags and compiler definition needed fo
Cmake script supplied by Gea
USE_FILE -> set to path to mo
package
Include directories -> adds B1
header search path
PROJECT -> (Cmake variable) points to directory of project in project command

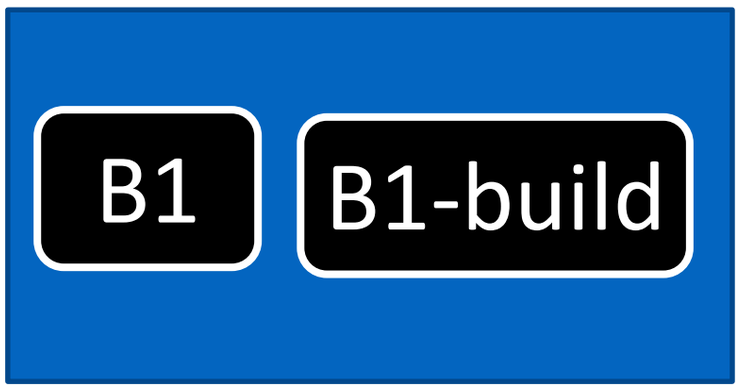
Build directory and Cmake for our 'task' application

1) If modifying the Geant4 examples, copy them to your $HOME first:

```
cp -r /usr/local/geant4/geant4.10.03.p02/examples/basic/B1 ~
```

2) Create a build directory*, where the compiled application will be put:

```
mkdir -p ~/B1-build
   cd ~/B1-build
```



*Note: It is possible (though not recommended) to compile **inside** source directory.

Run CMake

- In the build directory you just created, run CMake:

Path to Geant4

```
cmake -DGeant4_DIR=/usr/local/geant4/geant4.10.03.p03-install/lib64/Geant4-10.3.2/ ~/B1/
```

Path to source

```
-- The C compiler identification is GNU 4.8.5
-- The CXX compiler identification is GNU 4.8.5
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /path/to/build/directory
```

Compilation

- In the build directory, run make (and don't get a cup of coffee)
- You have only a couple of files, it should be ready in a minute or two
- An **executable** with the name of your application is created (e.g. exampleB1) in build directory
- **Macros** and other auxiliary files are copied into build directory

```
~/geant4-school-tasks/task1/task1-build % make
[  5%] Building CXX object CMakeFiles/Task.dir/Task.cc.o
[ 11%] Building CXX object CMakeFiles/Task.dir/src/DetectorConstruction.cc.o
[ 16%] Building CXX object CMakeFiles/Task.dir/src/DetectorMessenger.cc.o
[ 22%] Building CXX object CMakeFiles/Task.dir/src/G4LindhardPartition.cc.o
[ 27%] Building CXX object CMakeFiles/Task.dir/src/G4ScreenedNuclearRecoil.cc.o
[ 33%] Building CXX object CMakeFiles/Task.dir/src/PhysListEmStandard.cc.o
[ 38%] Building CXX object CMakeFiles/Task.dir/src/PhysListEmStandardNR.cc.o
[ 44%] Building CXX object CMakeFiles/Task.dir/src/PhysicsList.cc.o
[ 50%] Building CXX object CMakeFiles/Task.dir/src/PhysicsListMessenger.cc.o
[ 55%] Building CXX object CMakeFiles/Task.dir/src/PrimaryGeneratorAction.cc.o
[ 61%] Building CXX object CMakeFiles/Task.dir/src/PrimaryGeneratorMessenger.cc.o
[ 66%] Building CXX object CMakeFiles/Task.dir/src/RunAction.cc.o
[ 72%] Building CXX object CMakeFiles/Task.dir/src/StepMax.cc.o
[ 77%] Building CXX object CMakeFiles/Task.dir/src/StepMaxMessenger.cc.o
[ 83%] Building CXX object CMakeFiles/Task.dir/src/SteppingAction.cc.o
[ 88%] Building CXX object CMakeFiles/Task.dir/src/SteppingVerbose.cc.o
[ 94%] Building CXX object CMakeFiles/Task.dir/src/TrackingAction.cc.o
[100%] Linking CXX executable Task
[100%] Built target Task
~/geant4-school-tasks/task1/task1-build %
```

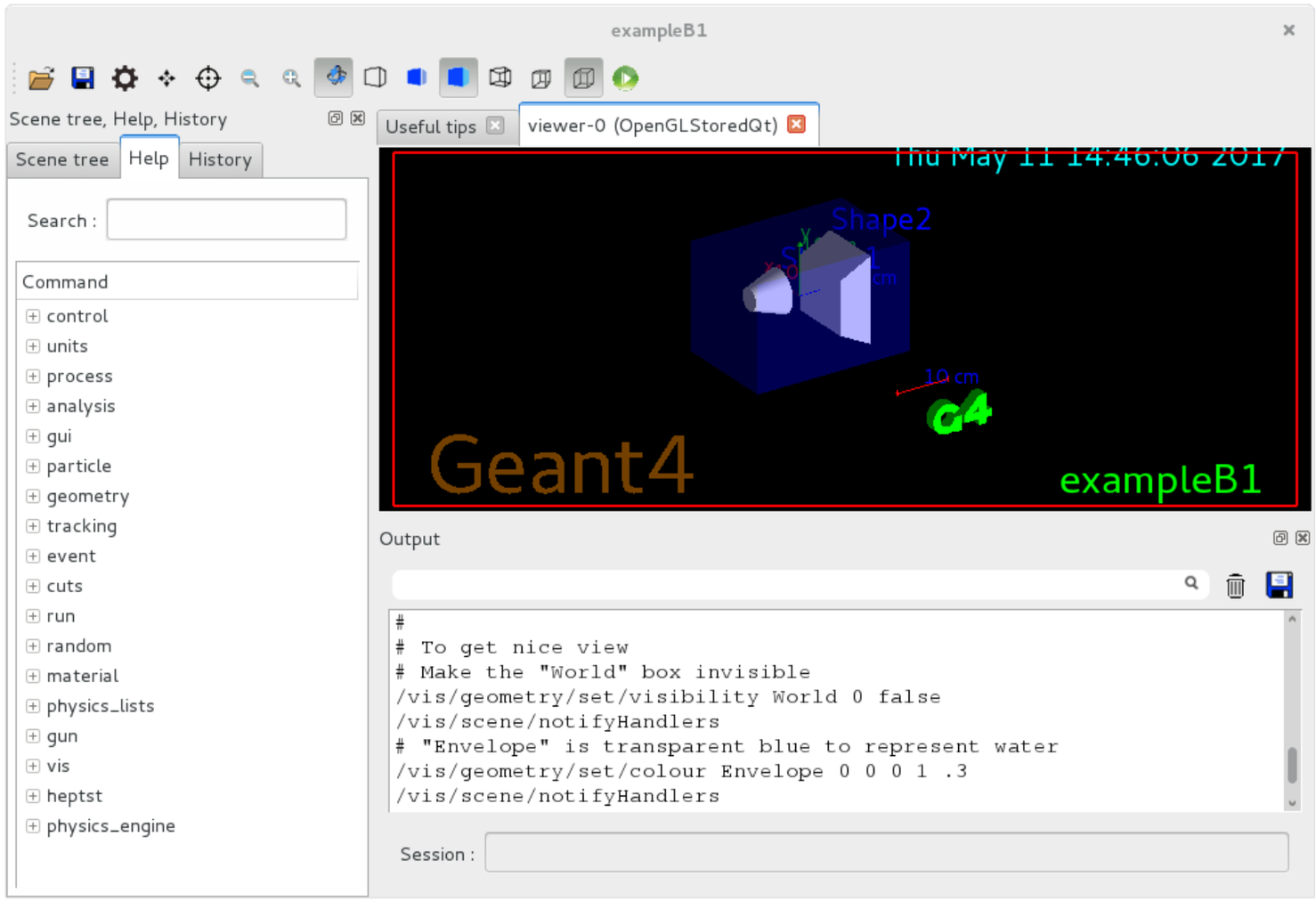
```
make -j2
```

Run the application - GUI

- Just type the name of your application, including the ./ identifier of current directory (e.g. ./exampleB1)

```
./exampleB1
```

```
Available UI session
types: [ Qt, GAG,
tcsh, csh ]
```



Thank you

Tasks to do:

- Exercise 0.1: Find and understand the GEANT4 environment file
- Exercise 0.2: Check your Geant4 environment
- Exercise 0.3: compile and run the basic example B1