Geant4 C++11/14/17 Hackathon: Getting Started

Ben Morgan, Ivana Hrivnacova

WARWICK
THE UNIVERSITY OF WARWICK

IJCLab
Irène Joliot-Curie
Laboratoire de Physique des 2 Infinis

Hackathon?

- *The goal of a hackathon is to create functioning software or hardware by the end of the event (according to Wikipedia!)*
  - *No pressure then!*
- Topics today are focussed on improving existing code/known new features using C++11/14/17
  - *So "creations" will be cleaner, simpler code in the CaTS topic, and design notes/prototypes for new Geometry/Touchables features utilizing C++17*
- An informal working session so all welcome to participate in the discussions, or to start up your own (and take breaks when you need)!
- **Organization/setup is virtual and somewhat ad-hoc, so bear with us, but we hope that it will prove useful, and maybe motivate further hackathons within Geant4 throughout the year when suitable topics arise.**

Today...

1. Brief review of material and tools that might be useful (~10-15min)
   a. *Making notes, C++ references, clang-tidy, IDEs*

2. Onto the topics: as we have two, we can either split into Zoom breakout rooms, or cover them one after the other (with an equal time for each).
   a. *What would you prefer?*
   b. *We can also use breakout rooms to discuss anything that comes up separately or in more detail - just let us know!*

3. Take 10min at the end to take stock of where we've got to, plus any feedback.

A quick review of tools/info that may help today…

- **All linked on the Indico page for this session**

Taking Notes/Developing Code

- Suggest the use of CERNs CodiMD instance: https://codimd.web.cern.ch
  - *Sign in with your usual CERN/gitlab account*
  - *Bit more flexible than Google Docs, especially for code!*
  - *We'll use a single doc whether we have breakout rooms or not, but keep sections apart!*
- Code development could use GitLab/Hub and existing or new repos at your discretion.
  - *Could use shared branches or Merge Requests to collaborate, or just have one "driver" (pair/N-programming)*
  - *If there are any VSCode users, could try the LiveShare functionality: https://code.visualstudio.com/learn/collaboration/live-share*

C++ References

- As we're focussing today on the use of modern C++ (up to C++17), our main references are:
  - *Geant4 C++11 guidelines (NB: JS is broken, so see Markdown WIP Here)*
  - *Geant4 Kernel, Examples Coding Guidelines*
    - *NB: New examples guidelines at Examples Parallel on Thursday*
  - *Geant4 Internal Seminar on C++11/14/17*
  - *Geant4 CODING_GUIDELINES.rst (additions/improvements welcome)*
  - *C++ Core Guidelines*
  - *CPP Reference*
  - *Compiler Explorer (in case you want to convince yourself that the modern C++ code is better/equivalent than the old!)*
- Any others?

The clang-tidy tool

- A useful tool to help identify code that can be improved for criteria of readability, modernity, and performance (among others)
  - *Not a silver bullet, but as we'll show, can get you a long way*
- Can be a little awkward to install on Linux, so don't worry if you don't have it or can't get it working today
  - *On Ubuntu 20 and newer:* `sudo apt install clang-tidy`
  - *On macOS/Homebrew:* `brew install llvm`
  - *From CVMFS/CentOS7 (?)*
- Quick demo on macOS…
- Fuller documentation in CODING_GUIDELINES.rst on GitLab

Let’s Hack!

… but keep it focussed on the topic at hand!