- **G4Run** (we will get back to this at the OptionalUserActions):
  - **G4Run** is a collection of **G4Event**-s (a **G4Event** is a collection of **G4Track**-s)
  - during a run, events are taken and processed one by one in an event-loop
  - **before the start of a run** i.e. at run initialisation (`G4RunManager::Initialize()`): the **geometry** is **constructed** and **physics** is **initialised**
  - **at the start of a run** (`G4RunManager::BeamOn()`): the **geometry is optimised** for tracking (voxelization), **physics tables are built**, then event processing starts i.e. entering into the event-loop
  - as log as the event processing is running, i.e. during the run, the user cannot modify **neither the geometry** (i.e. the detector setup) **nor the physics** settings
  - they **can be changed** though **between run**-s but the **G4RunManager** needs to be informed (re-optimise or re-construct geometry, re-build physics tables):
    - if the **geometry** has been changed, depending on the modifications:
      - `GeometryHasBeenModified()` re-voxelization but no re-`Construct`
      - `ReinitializeGeometry()` complete re-`Construct`

      or with the UI commands **/run/geometryModified** or **/run/reinitializeGeometry**
    - same for the **physics**: `PhysicsHasBeenModified()` or **/run/physicsModified**
  - we will get back to this when our application can run and produce information