

Introduction to Physics Modeling in Geant4

Oisín O'Connell (oisin@lanl.gov), ISR-1
Mentor: Dr. Mark Galassi

Los Alamos National Lab (LA-UR-20-25374)

August 5, 2020

Concepts

- ▶ Run manager
- ▶ Run
- ▶ Event
- ▶ Step
- ▶ Track
- ▶ User Actions

Inheritance, virtual functions

- ▶ Class derivations
- ▶ Self-derived, specify behavior
- ▶ Pass pointers- user initializations, actions
- ▶ Virtual functions

Example header file:

```
class BF3DetectorConstruction :  
    public G4VUserDetectorConstruction {  
    public:  
        BF3DetectorConstruction();  
  
        virtual G4VPhysicalVolume* Construct();  
};
```

Mandatory class derivations

- ▶ Pass w/SetUserInitialization
- ▶ G4VUserDetectorConstruction
- ▶ G4VUserActionInitialization
- ▶ G4VUserPhysicsList

Mandatory user initializations:

```
runManager->SetUserInitialization(MyDetectorConstruction);  
runManager->SetUserInitialization(MyPhysicsList);  
runManager->SetUserInitialization(MyActionInitialization);
```

Creating a world with objects

- ▶ G4VUserDetectorConstruction derivation
- ▶ "Construct()" virtual function- return root
- ▶ Self-defined or NIST manager materials
- ▶ Geometry, logical, physical volumes

Making a world geometry:

```
G4Box* world = new G4Box("World", hx, hy, hz);

G4LogicalVolume* logical_world =
    new G4LogicalVolume(world, // geometry
        NIST->FindOrBuildMaterial("G4_AIR")... // material

G4VPhysicalVolume* physical_world =
    new G4PVPlacement(0,
        G4ThreeVector(), // translation
        logical_world, // logical volume
        "world" //name..snip)
```

Physics lists

- ▶ Custom: G4VUserPhysicsList, processes, thresholds (cuts)
- ▶ Pre-defined: G4 reference lists
- ▶ This project: QGSP_BERT_HP
- ▶ Choose per project, naming conventions:
http://geant4.in2p3.fr/IMG/pdf_PhysicsLists.pdf

Using predefined physics lists:

```
G4VModularPhysicsList* physics_list =  
    new QGSP_BERT_HP;  
  
runManager->SetUserInitialization(physics_list);
```

Adding user actions

- ▶ G4VUserActionInitialization derivation
- ▶ Add user actions in derived class virtual functions
- ▶ Build(), BuildForMaster()
- ▶ Primary particle generation, run, track, step, etc

Adding user actions:

```
void NeutronActionInitialization::Build() const
{
    NeutronPrimaryGeneratorAction* npga =
        new NeutronPrimaryGeneratorAction;
    SetUserAction(npga);
    SetUserAction(new NeutronRunAction(npga));
}
```

Optional user action definitions and concepts

- ▶ G4V...Action
- ▶ UserPrimaryGenerator
- ▶ UserRun, UserEvent
- ▶ UserStacking, UserTracking
- ▶ UserStepping

Macros and our TUI

- ▶ G4UImanager- commands during execution
- ▶ Run simulations w/.mac, program, shell
- ▶ BeamOn

Running a simulation from the program:

```
G4UImanager* UI_manager = G4UImanager::GetUIpointer();  
UI_manager->ApplyCommand("/run/beamOn 1000")
```

Common problems and solutions

- ▶ Graphics options- set proper CMake variables
- ▶ Installing datasets- if auto-installing, consider manual wget instead:
\$CMAKE_INSTALL_PREFIX/share/myG4Version/data
- ▶ Post-installation environment variables: \$CMAKE_INSTALL_PREFIX/bin

Auto-installing datasets and using graphics drivers:

```
GEANT4_INSTALL_DATA      *OFF
..snip
GEANT4_USE_OPENGL_X11    *OFF
GEANT4_USE_[DRIVER]      *OFF
```

Oisin O'Connell (oisin@lanl.gov), ISR-1
Mentor: Dr. Mark Galassi