Środowisko programistyczne GEANT4

Leszek Adamczyk

Wydział Fizyki i Informatyki Stosowanej
Akademia Górniczo-Hutnicza

Wykłady w semestrze zimowym 2013/2014

Sterowanie symulacją

GEANT4 umożliwia sterowanie wykonaniem aplikacji na trzy sposoby:

- z poziomu kodu c++ (w programie main)
- za pomocą interwejsu użytkownika (UI)
  - z poziomu makr (poleceń UI zapisane w plikach tekstowych)

Pobranie wskaźnika do menegera UI

```
G4UImanager* UI = G4UImanager::GetUIpointer();
```

Przekazanie sterowania do UI

```
G4String macro = argv[1];
```

```
UI->ApplyCommand("/control/execute "+macro);
```

Uruchomienie programu

```
My run.mac
```

- interakcyjnie (poleceń UI przechwytywane przez interfejs)

Proszę wykonać ćwiczenie 1 z warsztatów nr 2

Interfejs użytkownika (UI)

- GEANT4 umożliwia prace z różnymi interfejsami:
  - znakowy(terminal tcsh);
  - z elementami graficznymi opartymi o widżety: Motif, Athena, Qt, Windows;
  - w pełni graficzny (GAG).
- Istnieje narzędzie G4UIExecutive które automatycznie uruchamia wybrany interfejs na podstawie zmiennych środowiskowych.
  Instancja nakładki na UI

```
G4UIExecutive * ui = new G4UIExecutive(argc,argv);
```

Przekazanie sterowania do UI

```
ui->SessionStart();
```

Uruchamiany jest pierwszy interfejs który ma zdefiniowana zmienną środowiskową:
`G4UI_USE_XX (XX= QT, XM, WIN32, GAG, TCSH)`

UI : polecenia

- Polecenie UI składa się z:
  /pełnej/ścieżki/dostępu/polecenia parametrów

```
/run/beamOn 3
```

- Parametry mogą być typu logicznego, liczbą całkowitą, rzeczywistą lub ”ciągiem znaków”;
- Parametry oddzielamy spacją;
- Parametry można omijać (przyjmowana jest wtedy wartość domyślna)
- Używamy ! gdy pierwszy parametr ma wartość domyślną a drugi zadaną

```
/dir/command ! 2
```

UI : polecenia systemowe

- Interfejs znakowy posiada wbudowane polecenia charakterystyczne dla Unix'a:
  - cd, pwd, ls
  - history – pokazuje poprzednio wykonane polecenia
  - !id – ponowne wykonanie polecenia o numerze id;
  - działaja klawisze strzałek i backspace (tylko tcsh)
  - help <polecenie> - opis i składnia polecenia
  - exit – zakończenie pracy
- Powyższe polecenia dostępne tylko interaktywnie, nie mogą być wykonane z poziomu programu w c++ ani wewnątrz makra.

UI : katalogi poleceń

```
Idle> ls
Command directory path : /
 Sub-directories :
   /control/    UI control commands.
   /units/      Available units.
   /process/   Process Table control commands.
   /persistency/   Control commands for Persistency package
   /geometry/   Geometry control commands.
   /tracking/   TrackingManager and SteppingManager control commands.
   /event/      EventManager control commands.
   /cuts/       Commands for G4VUserPhysicsList.
   /run/        Run control commands.
   /random/     Random number status control commands.
   /particle/   Particle control commands.
   /gun/        Particle Gun control commands.
   /material/   Commands for materials
   /gui/   UI interactors commands.
 Commands :
```

- po instancji menegera wizualizacji pojawi się katalog `/vis/`
- mozna zdefiniować własne polecenia np. w katalogu `/My/`
- link do opisu wbudowanych poleceń UI na stronie warsztatów.

UI : makra

- Makra są plikami zawierającymi polecenia UI
- Polecenia w makrze muszą posiadać pełną ścieżkę dostępu (brak poleceń cd, ...);
- # oznacza komentarz
- Makra można wykonać na dwa sposoby:
  - z poziomu interfejsu użytkownika lub z innego makra

```
/control/execute <nazwa_makra>
```

  - z poziomu kodu c++:

```
G4UImanager * UI = G4UImanager::GetUIpointer();
UI->ApplyCommand("/control/execute <nazwa_makra>");
```

UI : aliasy

- Aliasy w GEANT4 to nazwy zastępcze oraz zmienne makr;
- Aliasy definiujemy za pomocą polecenia //

```
/control/alias [nazwa]   [wartość]
```

np.

```
Idle> /control/alias  bo  "/run/beamOn"
```

- Wartość aliasu jest zawsze traktowana jak ciąg znaków;
- Alias jest automatycznie definiowany podczas wywołania poleceń

```
/control/loop
/control/foreach
```

- Alias może być używany jako składnik polecenia wywoływanego z poziomu UI lub wewnątrz makra. Do tego celu używamy nazwy aliasa w nawiasach pisanych np.

```
Idle> {bo}  3
```

UI : pętle

- Polecenia `/control/loop` i `/control/foreach` wykonują makro więcej niż raz:
  - `/control/loop [makro] [alias] [pocz] [końc] [kro`
    `[alias]` – alias którego wartość zmienia się przy każdym wywołaniu makra od `[pocz]` o `[krok]`

```
Idle> /control/loop run.mac Ekin 2. 10. 2.
```

  - `/control/foreach [makro] [alias] [lista]`
    `[alias]` – alias do kolejnych wartości z listy `[lista]`
    `[lista]` – musi być ograniczona ()

```
Idle> /control/foreach run.mac  Ekin "1. 7. 10."
```

- wewnątrz wywoływanego makra np.:
  `/gun/energy {Ekin} GeV`

Wizualizacje

GEANT4 umożliwia wizualizację:

- geometrii detektora;
- trajektorii czastek;
- depozytów energii;
- własnych obiektów (linie, symbole, opisy, ...)

Wizualizacje umożliwiają:

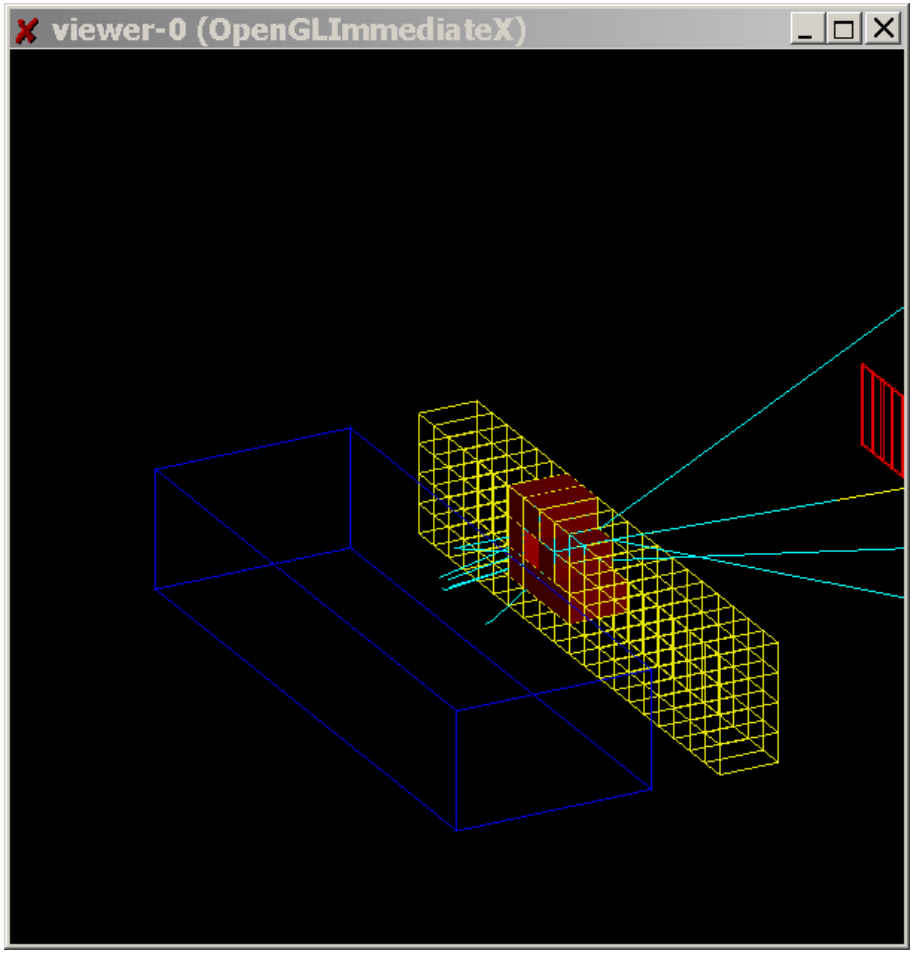
- tworzenie wysokiej jakości rysunków np. do publikacji;
- wykrywanie błędów w geometrii detektora (np. nakładanie się obszarów);
- interaktywne uzyskiwania informacji o wizualizowanych obiektach(pędy czastek, depozyty energii);

Wizualizacje: interfejsy

- GEANT4 posiada interfejsy obsługujące osiem zewnętrznych programów do wizualizacji:
  - OpenGl
  - HepRep
  - DAWN
  - OpenInventor
  - VRML
  - RayTracer
  - gMocren
  - ASCIITree
- Różne programy odpowiadają różnym potrzebom;
- Lista poleceń wydawanych z poziomu kodu c++ jest taka sama dla wszystkich programów.

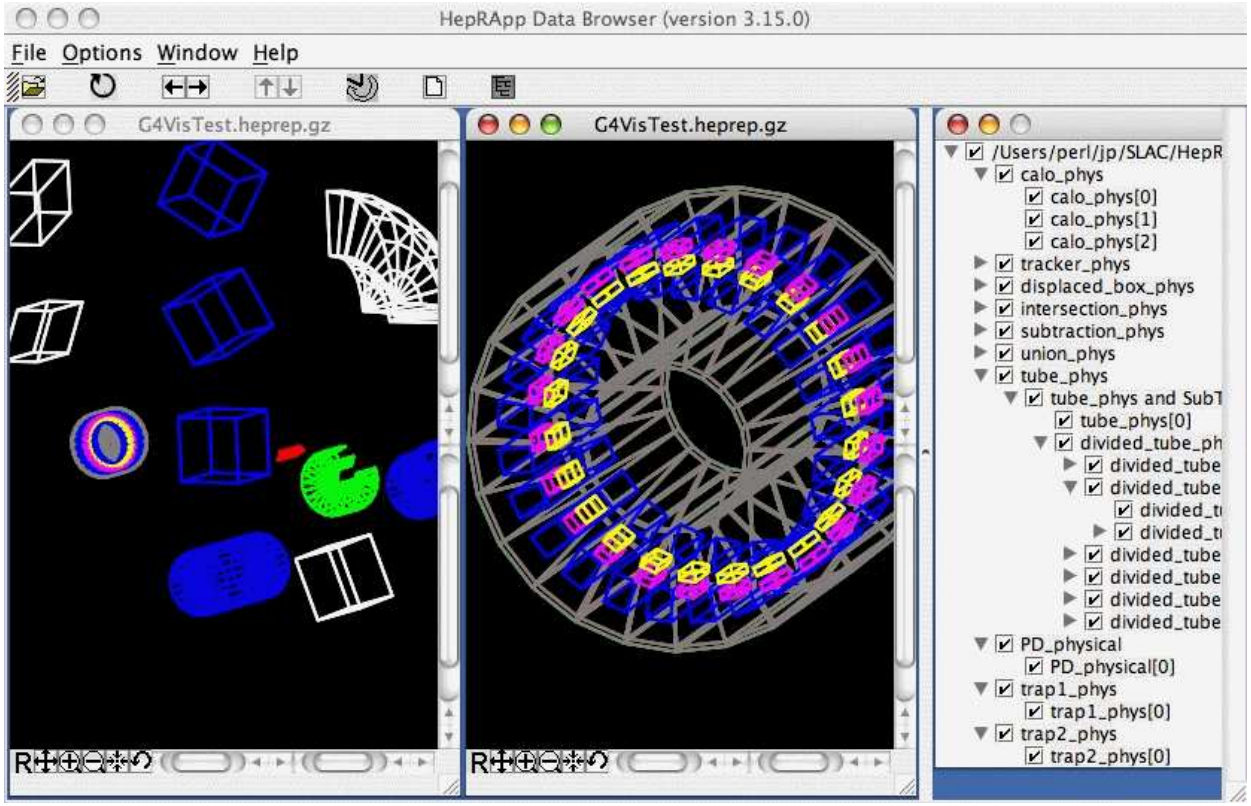
Wizualizacje: OpenGl

- Daje możliwość uzyskiwania szybkich wizualizacji: trajektorii, hitów, geometrii;
- Sterowanie tylko za pomocą poleceń interfejsu GEANT4;



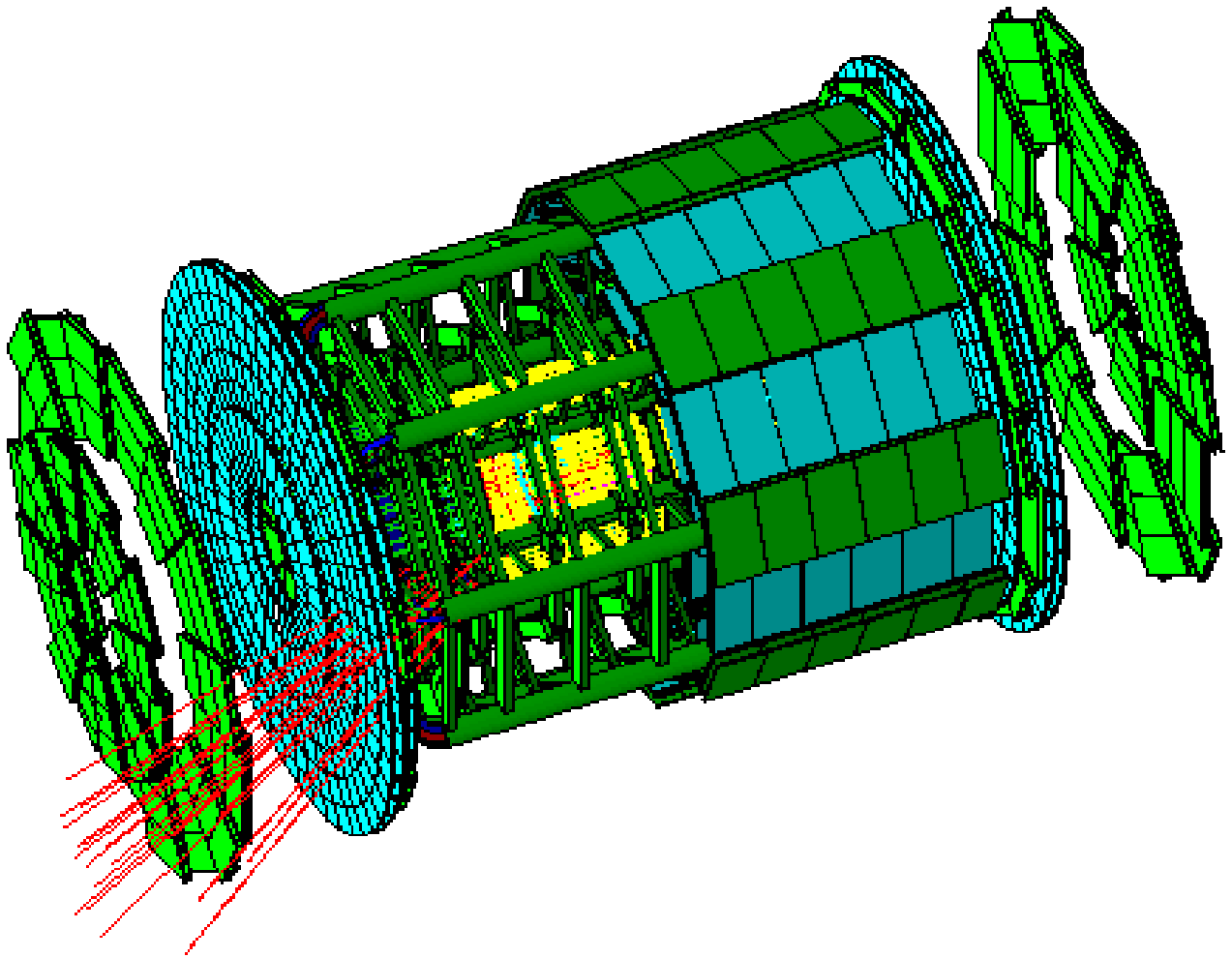
Wizualizacje: HepRep

- Daje możliwość interaktywnej (interfejs graficzny) wizualizacji: trajektorii, hitów, geometrii;
- Jest narzędziem do wykrywania błędów w geometrii detektora;
- Sterowanie za pomocą zewnętrznego interfejsu graficznego (wymaga tworzenia plików tymczasowych).



Wizualizacje: Dawn

- Daje możliwość wizualizacji: trajektorii, hitów, geometrii;
- Produkuje wysokiej jakości rysunki np. do publikacji
- Sterowanie za pomocą zewnętrznego programu (wymaga tworzenia plików tymczasowych).



Wizualizacje: interfejsy

- Sześć interfejsów niewymagających zewnętrznych bibliotek jest zawsze dostępnych w każdej instalacji GEANT4:
  - HepRep
  - DAWN
  - VRML
  - RayTracer
  - gMocren
  - ASCIITree
- Pozostałe wymagają instalacji bibliotek i ustawienia zmiennych środowiskowych podczas kompilacji jądra GEANT4;
  w szczególności OpenGL

Wizualizacje: interfejsy

- Niezależnie od interfejsu, kod źródłowy pozostaje w zasadzie taki sam;
- Wizualizacje realizujemy za pomocą: interfejsu użytkownika, z poziomu kodu c++ lub za pmocą zewnetrznych programów;
- Bezpośrednio z poziomu GEANT4 działają: OpenGL, OpenInventor, RayTracer i ASCIITree;
- Pozostałe, produkuje pliki umożliwiające wizualizacje za pomocą zewnętrznych aplikacji: HepRep, DAWN, VRML, gMocren;
- Można korzystać z kilku interfejsów w tym samym czasie (np. OpenGL/DAWN).

Wizualizacje: Manager

- Aby wizualizować należy utworzyć i zainicjalizować obiekt klasy wyprowadzonej z klasy G4VVisManager;
- Można skorzystać z gotowej implementacji G4VisExecutive;
- Użycie (w programie głównym):
  - Instancja

```
G4VisManager * visManager = new G4VisExecutive;
```

  - Inicjalizacja

```
visManager->Initialize();
```

  - Unicestwienie

```
delete visManager;
```

- Proszę wykonać ćwiczenie 3