

Geant4

Kısa Bir Tanıtım

Adnan Kılıç - adnank@uludag.edu.tr

Taylan Yetkin - tyetkin@yildiz.edu.tr

Ağırlıklı olarak linklerdeki konuşmaların slaytlarından yararlanılmıştır:

<https://agenda.infn.it/conferenceDisplay.py?confId=5699>

<https://hep.kisti.re.kr/indico/conferenceTimeTable.py?confId=51#20121029>

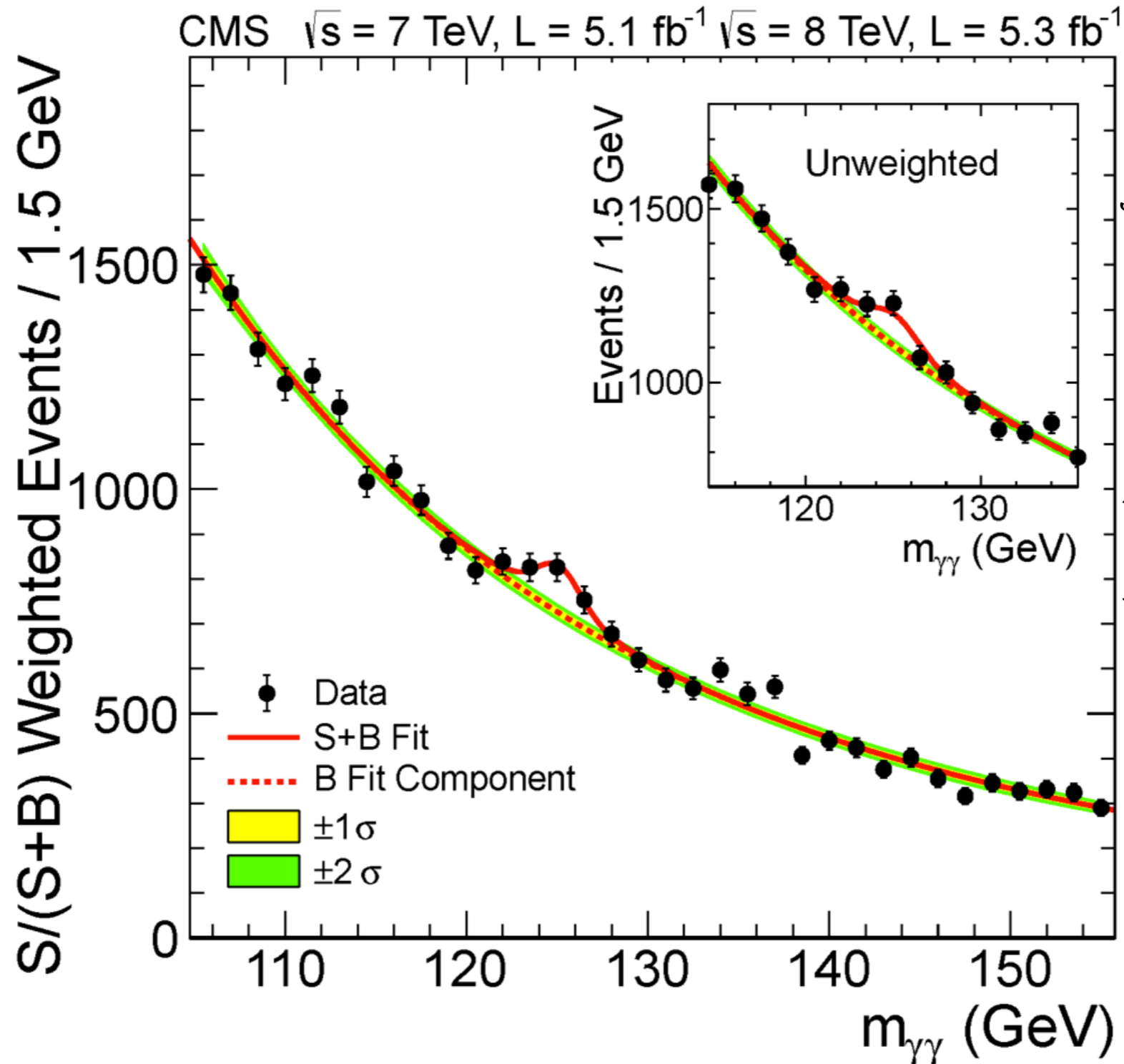
Önemli Not:

- Yaklaşık 1.5 saat sürecektir bir oturumda Geant4'ün sadece kısa bir tanıtımı yapılabilir.
 - ▶ Daha fazla kaynak kullanarak bu konudaki bilgilerinizi arttırmalısınız.
- Bu tanıtımdan sonra yaparak öğrenmeye çalışın, sorularınızı çekinmeden bizlere sorun.
- Aşağıdaki link küçük ödevler içermektedir, takip edip yapmaya çalışmanız iyi olacaktır:

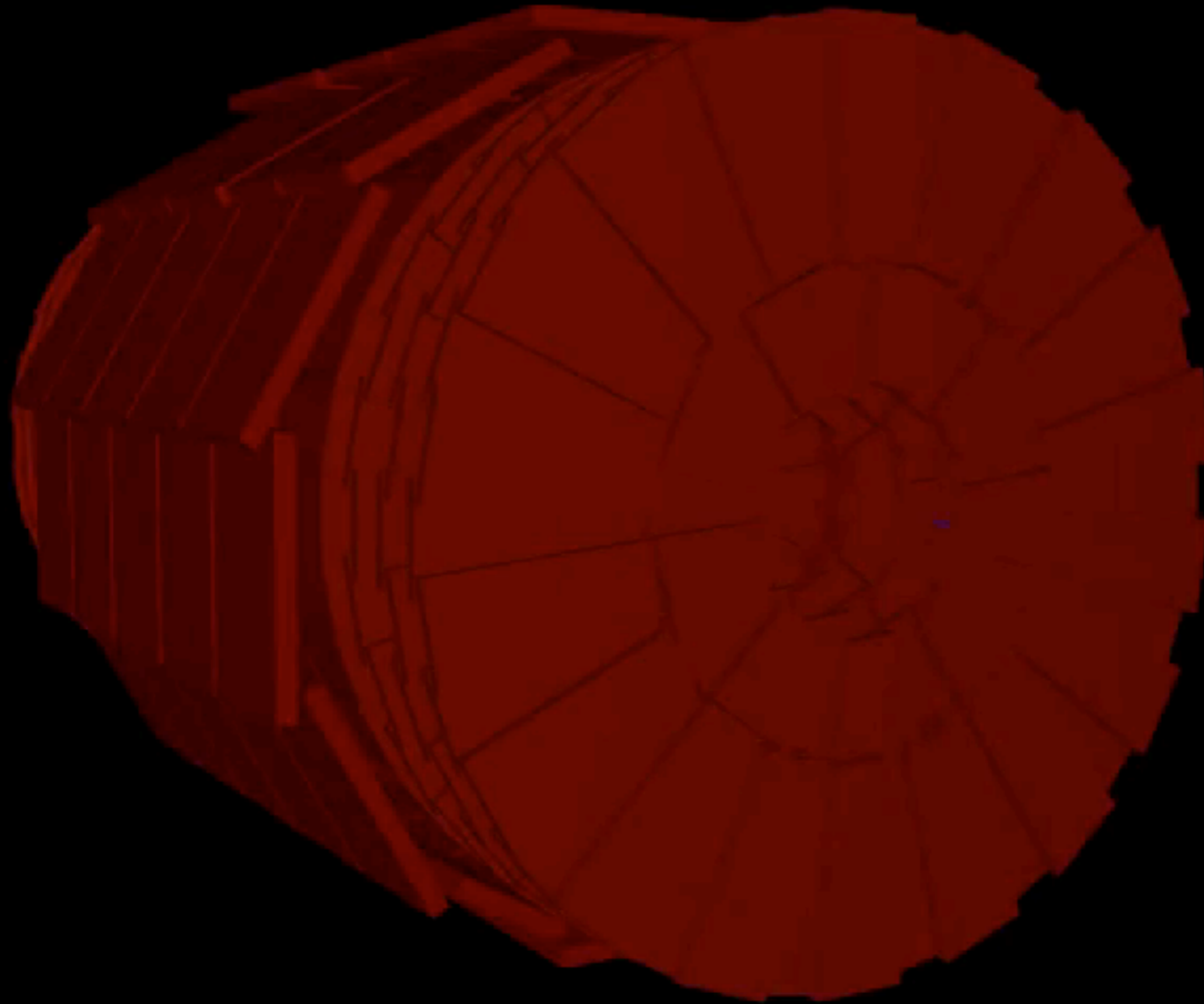
<http://geant4.lngs.infn.it/courseBelfast2013/exercises/introduction/index.html>

Giriş

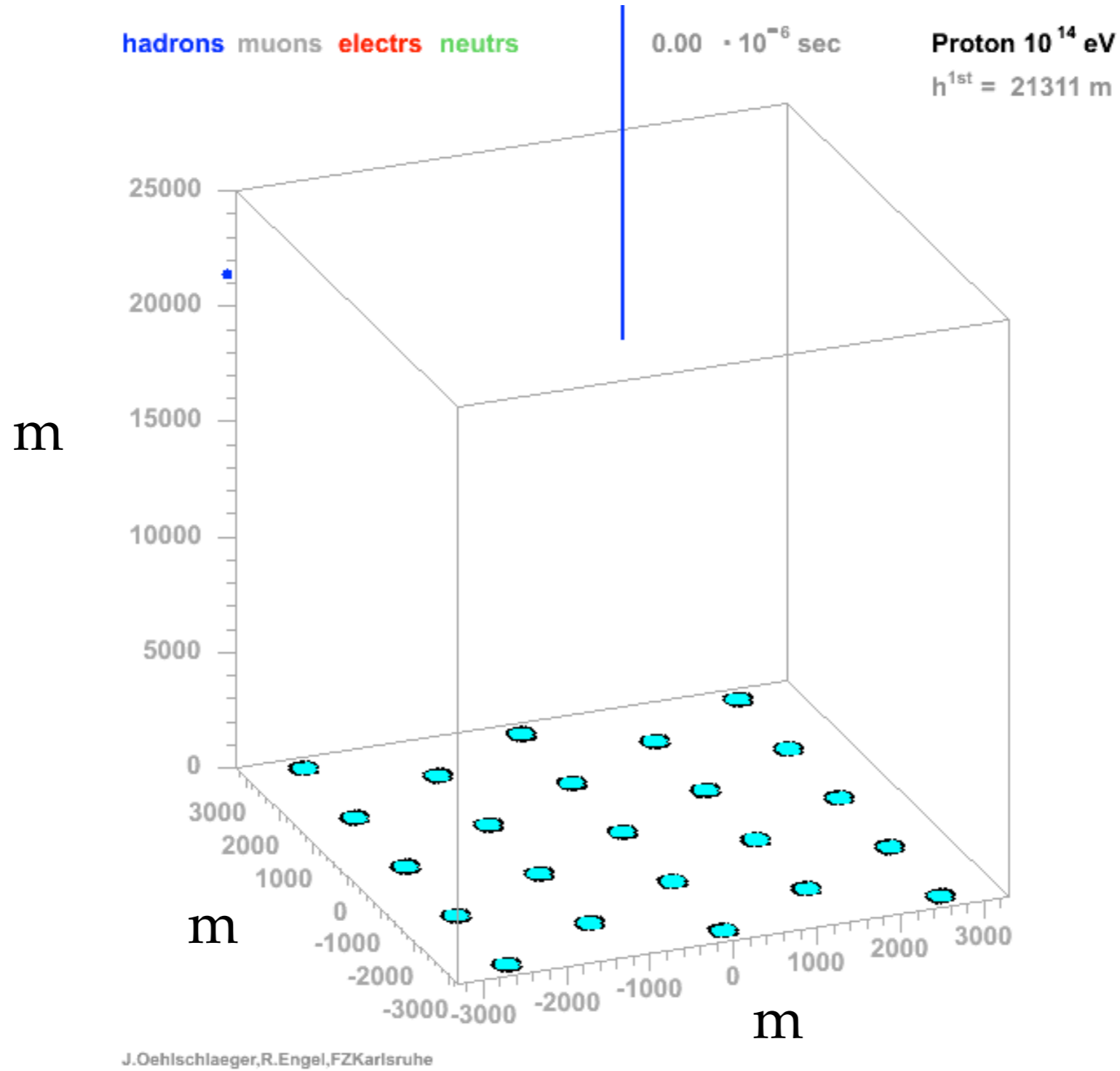
Geant4 Ne İşe Yarar?



Phys. Lett. B 716 (2012) 30-61



Birincil proton, havadaki sađanak (100 TeV)



kaynak: <http://www.lip.pt/~jespada/Research/PhysPAO.php>

Deney (Benzetim) İin Neye İhtiyacımız Var?

- Test edeceėimiz bir fikir/varsayım
- Varsayım ile ilgili fizik bilgisi
- Deney odası
- Algı
- Paracık kaynaėı
- Tetikleme mekanizması
- Veri toplama mekanizması
- Analiz/izim gereleri

Geant4:

- GEANT3'ten sonra gelen Geant4 YEF algıç benzetimlerinde kullanılan bir yazılım paketidir.
- Açık kaynak bir paket olup fizik süreçlerinin seçimi, GUI, görüntüleme, analiz paketleri tamamen kullanıcının seçimine bırakılmıştır.
- Nesne yönelimli programlama dil(ler)i ile yazılmıştır.
- Çeşitli fizik alanlarının ihtiyacına yanıt vermek için geliştirilmiştir: Ağır iyon fiziği, CP bozunumu, kozmik ışın fiziği, astrofizik, uzay fiziği ve medikal fizik.
- Madde ve parçacık etkileşimlerinin benzetim paketi olan G4 YEF'te çok önemli keşiflerde kullanılmıştır (Higgs bozonu).

Geant4

<http://geant4.cern.ch>

Geant 4

[Download](#) | [User Forum](#) | [Gallery](#)
[Contact Us](#)

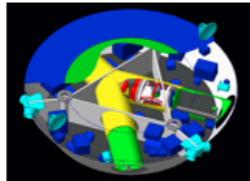
Geant4 is a toolkit for the simulation of the passage of particles through matter. Its areas of application include high energy, nuclear and accelerator physics, as well as studies in medical and space science. The two main reference papers for Geant4 are published in *Nuclear Instruments and Methods in Physics Research A* [506 \(2003\) 250-303](#), and *IEEE Transactions on Nuclear Science* [53 No. 1 \(2006\) 270-278](#).

Applications



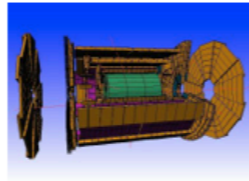
A [sampling of applications](#), technology transfer and other uses of Geant4

User Support



[Getting started](#), [guides](#) and information for users and developers

Publications



[Validation of Geant4](#), results from experiments and publications

Collaboration



[Who we are](#): collaborating institutions, [members](#), organization and legal information

News

- 6 December 2013 - **Release 10.0** is available from the [Download](#) area.
- 24 May 2013 - **Patch-02 to release 9.6** is available from the [download](#) area.

Events

- [SLAC Geant4 Tutorial Course](#), Jen-Hsun Huang Engineering Center, Stanford (US), **3-6 March 2014**.
- 10th Geant4 Space Users Workshop, at NASA/MSFC, Huntsville, Alabama (USA), **27-29 May 2014**.
- [International Workshop on Monte Carlo Techniques in Medical Physics](#), Quebec City (Canada), **17-20 June 2014**.
- 19th Geant4 Collaboration Meeting, Okinawa (Japan), **29 September - 4 October 2014**.
- [Past events](#)

Kullanıcı Desteđi

<http://geant4.web.cern.ch/geant4/support/index.shtml>

Geant 4

[Download](#) | [User Forum](#) | [Gallery](#)
[Contact Us](#)

Search Geant4

[Home](#) > [User Support](#)

User Support

1. [Getting started](#)
2. [Training courses and materials](#)
3. Source code
 - a. [Download page](#)
 - b. [LXR code browser](#) -or- draft [doxygen documentation](#)
4. [Frequently Asked Questions \(FAQ\)](#)
5. [Bug reports and fixes](#)
6. [User requirements tracker](#)
7. [User Forum](#)
8. [Documentation](#)
 - a. [Introduction to Geant4](#)
 - b. [Installation Guide](#)
 - c. [Application Developers Guide](#)
 - d. [Toolkit Developers Guide](#)
 - e. [Physics Reference Manual](#)
9. [Examples](#)
10. Physics lists
 - a. [Electromagnetic](#)
 - b. [Hadronic](#)
11. User Aids
 - a. [Tips for improving CPU performance](#)
 - b. [Process/model catalog](#)
 - c. [General particle source manual](#)
12. [Contact Coordinators & Contact Persons](#)

Related Links

- [Object Oriented Analysis & Design](#)
- [Archive](#)
- [Mailing list subscription](#)
- [User requirements document \(pdf\)](#)
- [Technical Forum](#)

Kullanıcı Desteđi

<http://geant4.web.cern.ch/geant4/support/index.shtml>

Geant 4 [Download](#) [User Forum](#) [Gallery](#)
[Contact Us](#)

[Home](#) > [User Support](#)

User Support

- [Getting started](#)
- [Training courses and materials](#)
- Source code
 - [Download page](#)
 - [LXR code browser](#) -or- draft [doxygen documentation](#)
- [Frequently Asked Questions \(FAQ\)](#)
- [Bug reports and fixes](#)
- [User requirements tracker](#)
- [User Forum](#)
- Documentation
 - [Introduction to Geant4](#)
 - [Installation Guide](#)
 - [Application Developers Guide](#)
 - [Toolkit Developers Guide](#)
 - [Physics Reference Manual](#)
- [Examples](#)
- Physics lists
 - [Electromagnetic](#)
 - [Hadronic](#)
- User Aids
 - [Tips for improving CPU performance](#)
 - [Process/model catalog](#)
 - [General particle source manual](#)
- [Contact Coordinators & Contact Persons](#)

Related Links

- [Object Oriented Analysis & Design](#)
- [Archive](#)
- [Mailing list subscription](#)
- [User requirements document \(pdf\)](#)
- [Technical Forum](#)

Kullanıcı Forumu

<http://hypernews.slac.stanford.edu/HyperNews/geant4/cindex>

GEANT4 at hypernews.slac.stanford.edu Forum List by Category

Not Logged In ([login](#))

Geant 4

[Forums by Category](#)
[Forums by Time Order](#)
[Request a New Forum](#)

[Recent Postings](#)
[Search in Forums](#)
[Subscribe to Forums](#)

[Member Info](#)
[Members List](#)
[New Member](#)

[Overview](#)
[Contact Admin](#)

[Page Help](#)

Category: Applications

[Educational Applications](#) [Industrial instruments](#) [Medical Applications](#) [Space Applications](#)

Category: Control of runs, events, tracks, particles

[Event and Track Management](#) [Multithreading](#) [Particles](#) [Run Management](#)

Category: Experimental Setup

[Biasing and Scoring](#) [Fields: Magnetic and Otherwise](#) [Geometry](#) [Hits, Digitization and Pileup](#)

Category: General matters

[Documentation and Examples](#) [HyperNews System Announcements](#) [Hypernews Testing](#) [Installation and Configuration](#) [User Requirements](#)

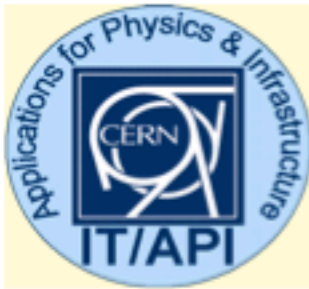
Category: Interfaces

[\(Graphical\) User Interfaces](#) [Analysis](#) [Persistency](#) [Visualization](#)

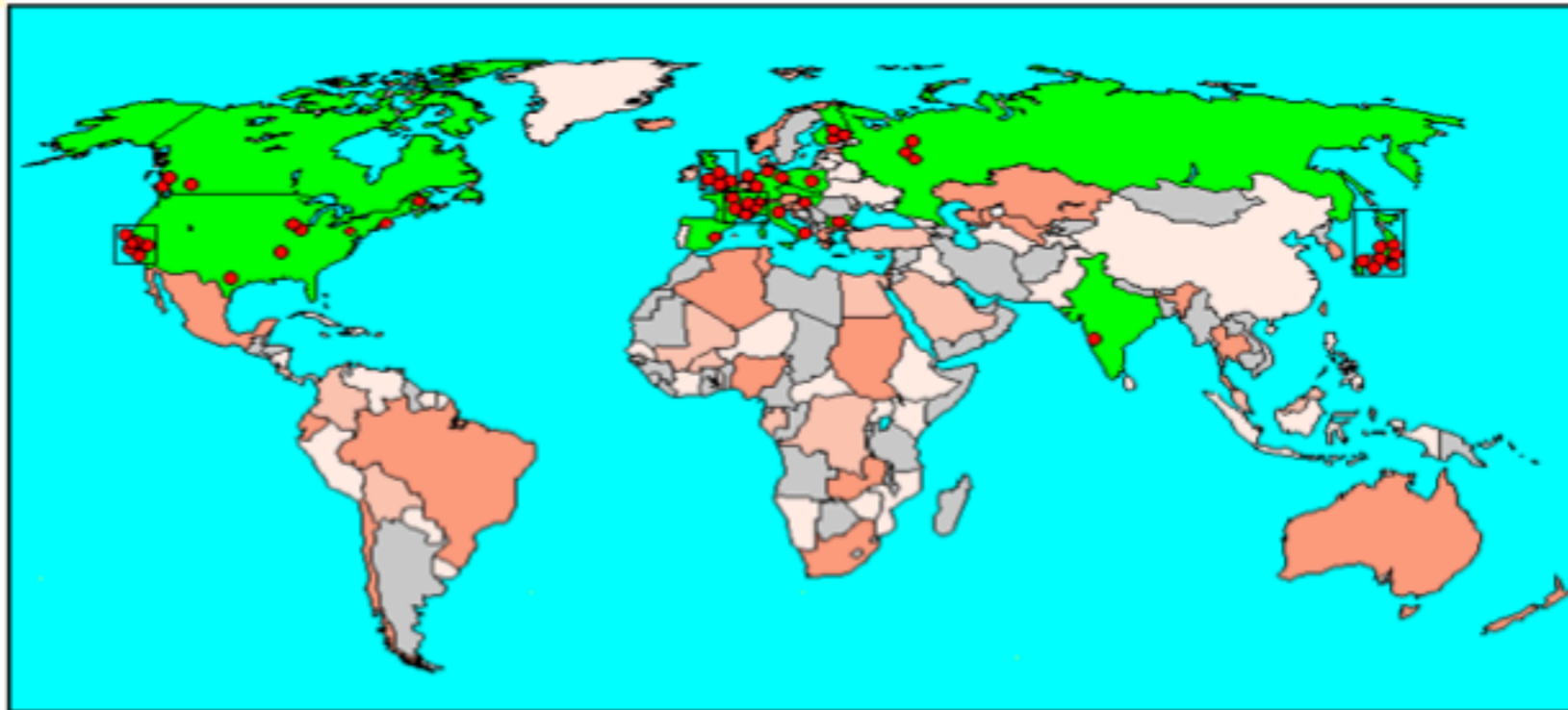
Category: Physics

[Biasing and Scoring](#) [Electromagnetic Processes](#) [Fast Simulation, Transportation & Others](#) [Hadronic Processes](#) [Physics List](#) [Processes Involving Optical Photons](#)

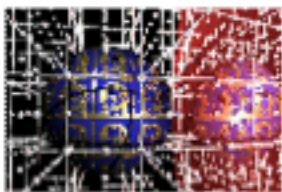
Geant4 İşbirliği



TRIUMF



Lebedev



J.W.Goethe
Universität



Collaborators also from non-member institutions, including
Budker Inst. of Physics
IHEP Protvino
MEPHI Moscow
Pittsburg University
Northeastern University
Wollongong University

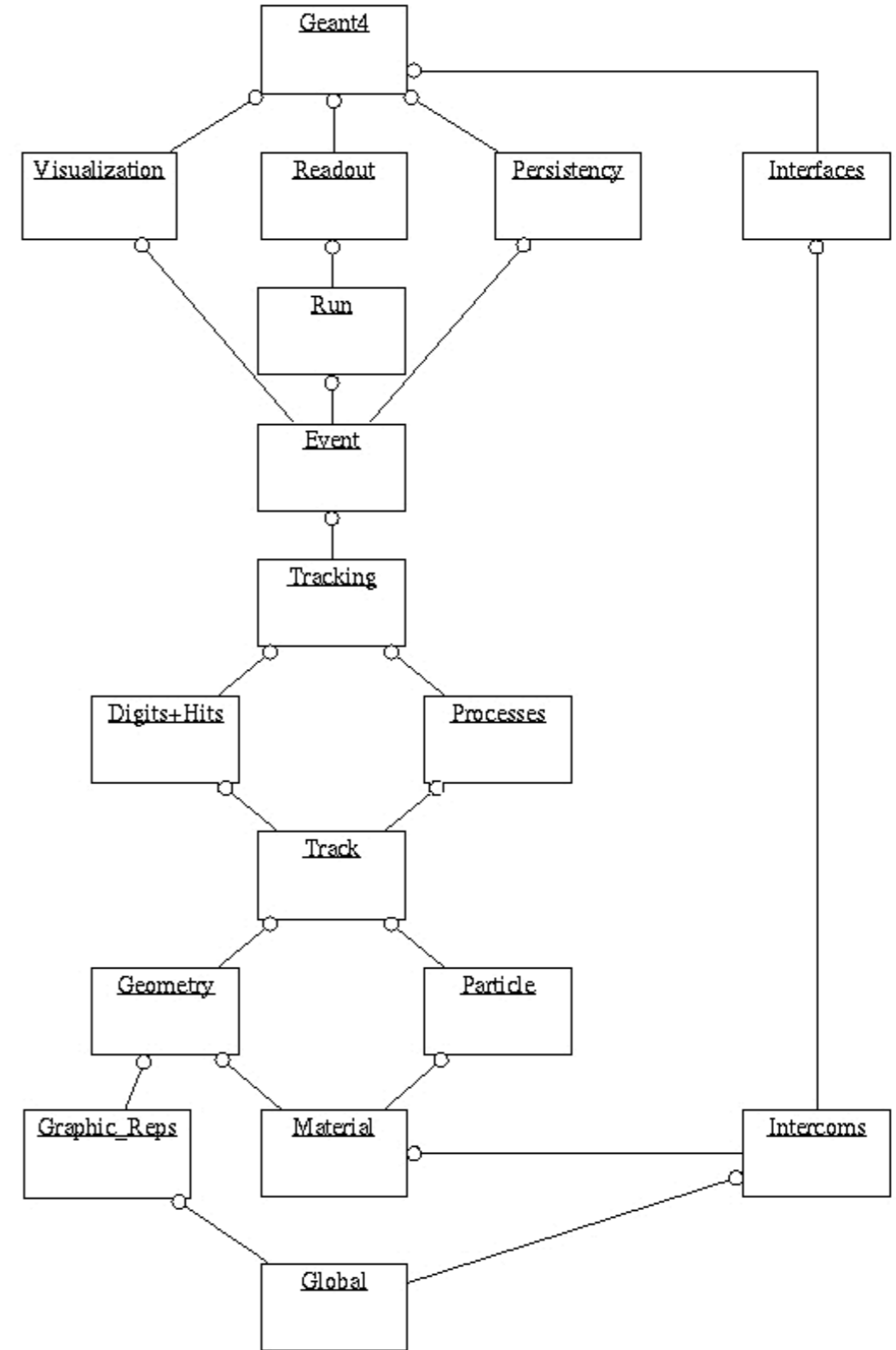
Terminoloji/Meslek Argosu

- Koşmak (run), olay (event), iz (track), adım (step), adım noktası (step point)
- iz \Leftrightarrow gezinge (trajectory), adım \Leftrightarrow gezinge noktası
- Süreç (process)
 - ▶ durağan (at rest), adım boyunca (along step), adımdan sonra (post step)
- Kesme (Cut) = üretme eşiği (production threshold)
- Hassas algıç (sensitive detector), hesap (score), vuru (hit), vuru topluluğu (hits collection)

Kısa Bir Geant4 Gezintisi

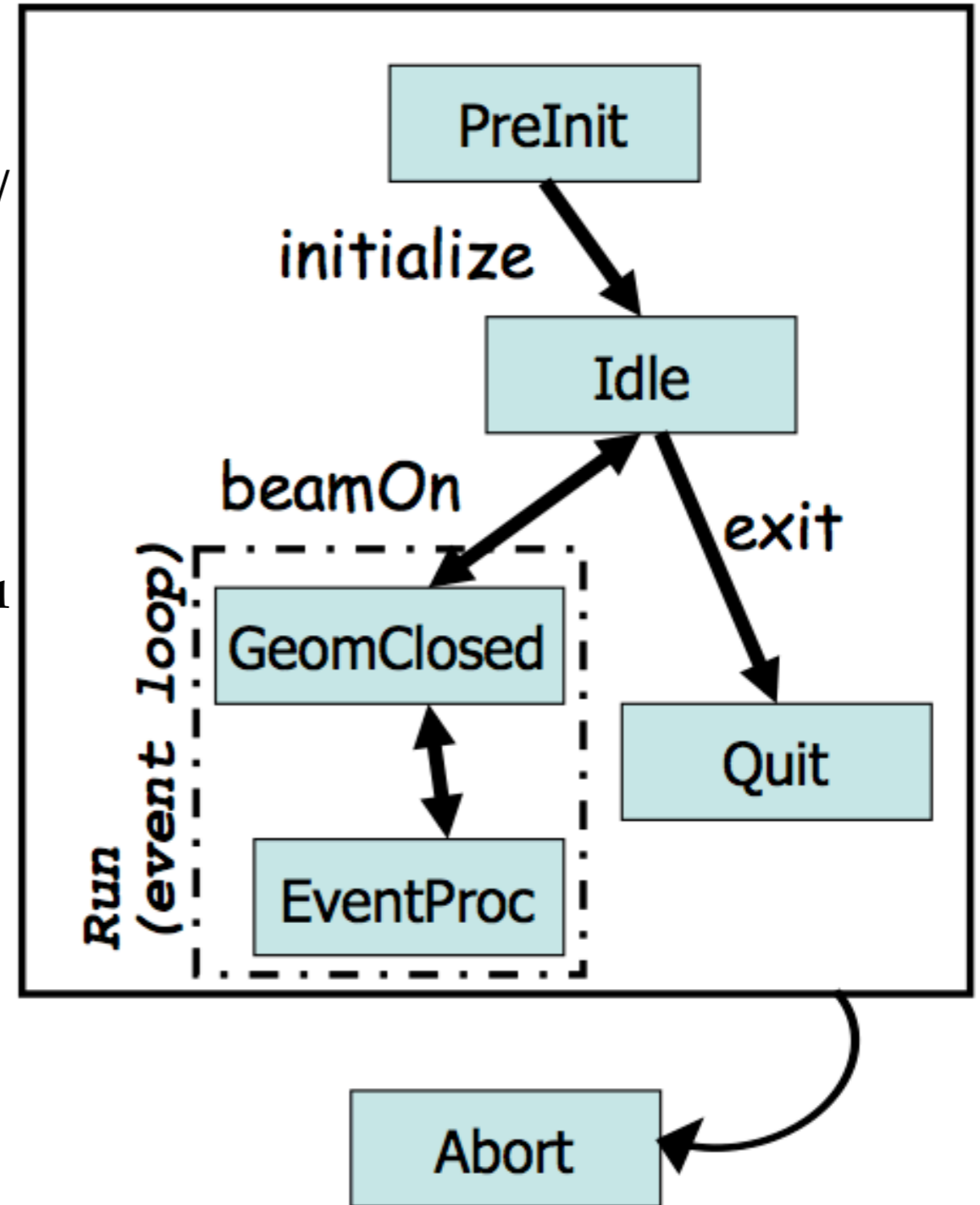
Sınıf Bölümleme (Class Category Diagram)

- Geant4 17 bölümden oluşur.
 - ▶ Her bölüm bağımsız olarak sorumlu ÇG tarafından geliştirilir ve bakımı yapılır.
 - ▶ Bölümler arasındaki ayayüzler en tepedeki ÇG tarafından düzenlenir.
- Geant4 Çekirdeği
 - ▶ koşu, olay, iz, adım, vuru, gezinge (run, event, track, step, hit, trajectory) yi işler.
 - ▶ Geometrik betimlemeler ve fizik süreçleri için taslaklar sunar.

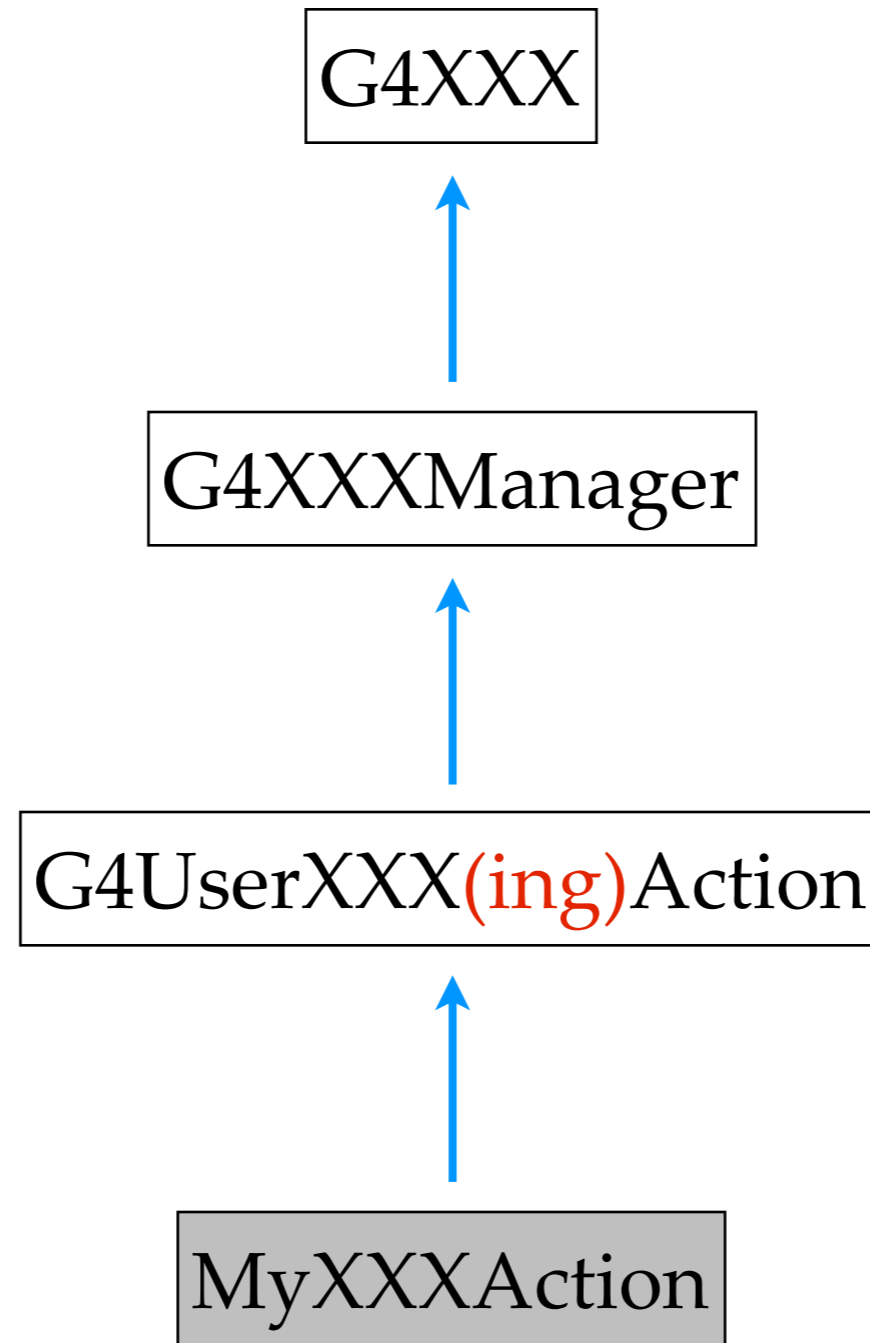


Bir Durum Makinesi Olarak Geant4

- Geant4 altı tane uygulama durumuna sahiptir
 - ▶ G4State_PreInit
 - Malzeme, Geometri, Parçacık ve/veya Fizik Süreçleri başlamaya hazırlanmalı/tanımlanmalıdır
 - ▶ G4State_Idle
 - Bir koşuyu başlatmaya hazır
 - ▶ G4State_GeomClosed
 - Geometri optimize edilmiş ve bir olayı işlemeye hazır hale getirilmiştir
 - ▶ G4State_EventProc
 - Bir olay işlenmektedir
 - ▶ G4State_Quit
 - (Normal) sonlandırma
 - ▶ G4State_Abort
 - Onulmaz bir hata olmuştur ve program yarıda kesilip çıkılır



Sıklıkla Karşılaştığınız Tipik Yapı



XXX: Run, Event, Track, Step, ...

Geant4'te Fizik

- Çok sayıdaki parçacığı ve enerji aralıklarını kapsayan tek bir fizik modeli geliştirmek gerçekçi/mümkün değildir.
 - Bunun yerine kuramsal çalışmalardan, parametrelendirilmiş ya da deney sonuçlarına dayanan bağıntıların/ilişkilerin bir karışımı ile fiziğin bu geniş alanı ele alınabilir. Tesir kesitleri ve fizik modelleri seçilmiş olan bir sürece keyfi olarak bağlanabilirler (polymorphism mechanism)
- Geant4'ün sunmuş olduğu süreçler:
 - ▶ elektromanyetik
 - ▶ hadronik
 - ▶ foton/lepton-hadron
 - ▶ optik foton
 - ▶ bozunum
 - ▶ sağanak (shower) oluşumu

Geant4'un Koşulması

- Gerçek bir deneye benzer olarak Geant4'ün koşturmayaya (çalışmaya) başlaması "Beam On" (Demet Geliyor) ile olur.
- Koşu süresince kullanıcı
 - ▶ algıç düzenlenmesini
 - ▶ fiziksel süreçlerin ayarlarını değiştiremez.
- Kavramsal olarak bir koşu aynı algıç ve fizik koşullarını paylaşan olaylar topluluğudur.
 - ▶ Bir koşu olay halkasından oluşur.
- Bir koşunun başında geometri yön bulma için en uygun biçime getirilir ve tesir kesitleri geometride verilen malzemeler ve kesme değeri kullanılarak hesaplanır.
- **G4RunManager** sınıfı bir koşuyu yönetir, koşu **G4Run** sınıfı ya da bu sınıftan kullanıcı tarafından türetilmiş bir sınıfla temsil edilir.
 - ▶ Bir koşu sınıfı koşunun sonucunda özet sonuçları içerebilir.
- **G4UserRunAction** sınıfı ise kullanıcıların tercih edebileceği bir sınıftır.

G4Run

```
30 #ifndef G4Run_h
31 #define G4Run_h 1
32
33 #include "globals.hh"
34 #include <vector>
35 class G4Event;
36 class G4HCTable;
37 class G4DCtable;
38
39 // class description:
40 //
41 // This class represents a run. An object of this class is constructed
42 // and deleted by G4RunManager. Basically the user should use only the
43 // get methods. All properties are set by G4RunManager.
44 //
45
46 class G4Run
47 {
48 public:
49     G4Run();
50     virtual ~G4Run();
51
52 private:
53     // These copy constructor and = operator must not be used.
54     G4Run(const G4Run &) {}
55     G4Run& operator=(const G4Run &) { return *this; }
56
57     public: // with description
58     inline G4int GetRunID() const
59     { return runID; }
60     // Returns the run ID. Run ID is set by G4RunManager.
61     inline G4int GetNumberOfEvent() const
62     { return numberOfEvent; }
63     // Returns number of events processed in this run. The number is
64     // incremented at the end of each event processing.
65     inline G4int GetNumberOfEventToBeProcessed() const
66     { return numberOfEventToBeProcessed; }
67     inline const G4HCTable* GetHCTable() const
68     { return HCTable; }
69     // List of names of hits collection
70     inline const G4DCtable* GetDCtable() const
71     { return DCtable; }
72     // List of names of digi collection
73     inline const G4String& GetRandomNumberStatus() const
74     { return randomNumberStatus; }
75     // Return random number status at the beginning of this run
```

G4UserRunAction

G4UserRunAction.hh

```
30 #ifndef G4UserRunAction_h
31 #define G4UserRunAction_h 1
32
33 class G4Run;
34
35 // class description:
36 //
37 // This is the base class of a user's action class which defines the
38 // user's action at the beginning and the end of each run. The user can
39 // override the following two methods but the user should not change
40 // any of the contents of G4Run object.
41 // virtual void BeginOfRunAction(const G4Run* aRun);
42 // virtual void EndOfRunAction(const G4Run* aRun);
43 // The user can override the following method to instantiate his/her own
44 // concrete Run class. G4Run has a virtual method RecordEvent, so that
45 // the user can store any information useful to him/her with event statistics.
46 // virtual G4Run* GenerateRun();
47 // The user's concrete class derived from this class must be set to
48 // G4RunManager via G4RunManager::SetUserAction() method.
49 //
50 #include "G4Types.hh"
51
52 class G4UserRunAction
53 {
54 public:
55     G4UserRunAction();
56     virtual ~G4UserRunAction();
57
58 public:
59     virtual G4Run* GenerateRun();
60     virtual void BeginOfRunAction(const G4Run* aRun);
61     virtual void EndOfRunAction(const G4Run* aRun);
62
63 protected:
64     G4bool isMaster;
65
66 public:
67     inline void SetMaster(G4bool val=true)
68     { isMaster = val; }
69     inline G4bool IsMaster() const
70     { return isMaster; }
71 };
72
73 #endif

```

extended/analysis/shared/

```
37 #ifndef RunAction_h
38 #define RunAction_h 1
39
40 #include "G4UserRunAction.hh"
41 #include "globals.hh"
42
43 //.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....
44
45 class G4Run;
46 class HistoManager;
47
48 class RunAction : public G4UserRunAction
49 {
50 public:
51     RunAction(HistoManager*);
52     virtual ~RunAction();
53
54     virtual void BeginOfRunAction(const G4Run*);
55     virtual void EndOfRunAction(const G4Run*);
56
57     void fillPerEvent(G4double, G4double, G4double, G4double);
58
59 private:
60     HistoManager* fHistoManager;
61
62     G4double fSumEAbs, fSum2EAbs;
63     G4double fSumEGap, fSum2EGap;
64
65     G4double fSumLABs, fSum2LABs;
66     G4double fSumLGap, fSum2LGap;
67 };
68
69 //.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....
70
71 #endif

```

Geant4'te Olay

- Olay Geant4'te benzetimin temel ünitesidir.
- İşleme sürecinin başında, birincil izler (primary tracks) üretilir ve bunlar yığına (stack) konulur.
- Yığındaki izler birer birer alınır ve “izlenir”. Ortaya çıkan ikincil izler de yığına itilir.
 - ▶ “İzleme” yığınlardaki izler bitene kadar devam eder.
- Yığın boşaldığında bir olayın işlenmesi sona erer.
- **G4Event** sınıfı bir olayı temsil eder. İşlemenin sonucunda şu nesnelere sahiptir:
 - ▶ Birincil köşelerin (primary vertices) ve parçacıkların listesi (girdi olarak)
 - ▶ Vuru ve gezinge toplulukları (çıktı olarak)
- **G4EventManager** sınıfı bir olayın işlenmesinden sorumludur.
- **G4UserEventAction** sınıfı ile kullanıcılar bu sınıfa ulaşıp işlem yapabilirler.

G4Event

```
30 // class description:
31 //
32 // This is the class which represents an event. A G4Event is constructed and
33 // deleted by G4RunManager (or its derived class). When a G4Event object is
34 // passed to G4EventManager, G4Event must have one or more primary vertexes
35 // and primary particle(s) associated to the vertex(es) as an input of
36 // simulating an event.
37 // G4Event has trajectories, hits collections, and/or digi collections.
38
39 #ifndef G4Event_h
40 #define G4Event_h 1
41
42 #include "globals.hh"
43 #include "evtdefs.hh"
44 #include "G4Allocator.hh"
45 #include "G4PrimaryVertex.hh"
46 #include "G4HCofThisEvent.hh"
47 #include "G4DCofThisEvent.hh"
48 #include "G4TrajectoryContainer.hh"
49 #include "G4VUserEventInformation.hh"
50
51 class G4VHitsCollection;
52 class G4Event
53 {
54 public:
55     G4Event();
56     G4Event(G4int evID);
57     ~G4Event();

```

```
139 public: // with description
140     inline G4int GetEventID() const
141     { return eventID; }
142     // Returns the event ID
143     inline void AddPrimaryVertex(G4PrimaryVertex* aPrimaryVertex)
144     {
145         if( thePrimaryVertex == 0 )
146             { thePrimaryVertex = aPrimaryVertex; }
147         else
148             { thePrimaryVertex->SetNext( aPrimaryVertex ); }
149         numberOfPrimaryVertex++;
150     }
151     // This method sets a new primary vertex. This method must be invoked
152     // exclusively by G4VPrimaryGenerator concrete class.
153     inline G4int GetNumberOfPrimaryVertex() const
154     { return numberOfPrimaryVertex; }
155     // Returns number of primary vertexes the G4Event object has.
156     inline G4PrimaryVertex* GetPrimaryVertex(G4int i=0) const
157     {
158         if( i == 0 )
159             { return thePrimaryVertex; }
160         else if( i > 0 && i < numberOfPrimaryVertex )
161             {
162                 G4PrimaryVertex* primaryVertex = thePrimaryVertex;
163                 for( G4int j=0; j<i; j++ )
164                     {
165                         if( primaryVertex == 0 ) return 0;
166                         primaryVertex = primaryVertex->GetNext();
167                     }
168                 return primaryVertex;
169             }
170         else
171             { return 0; }
172     }
173     // Returns i-th primary vertex of the event.
174     inline G4HCofThisEvent* GetHCofThisEvent() const
175     { return HC; }
176     inline G4DCofThisEvent* GetDCofThisEvent() const
177     { return DC; }
178     inline G4TrajectoryContainer* GetTrajectoryContainer() const
179     { return trajectoryContainer; }
180     // These three methods returns the pointers to the G4HCofThisEvent
181     // (hits collections of this event), G4DCofThisEvent (digi collections
182     // of this event), and G4TrajectoryContainer (trajectory container),
183     // respectively.
```


G4UserEventAction

```
32 #ifndef G4UserEventAction_h
33 #define G4UserEventAction_h 1
34
35 class G4EventManager;
36 class G4Event;
37
38 // class description:
39 //
40 // This is the base class of one of the user's optional action classes.
41 // The two methods BeginOfEventAction() and EndOfEventAction() are invoked
42 // at the beginning and the end of one event processing. These methods are
43 // invoked by G4EventManager.
44 // Be aware that BeginOfEventAction() is invoked when a G4Event object is
45 // sent to G4EventManager. Thus the primary vertexes/particles have already
46 // been made by the primary generator. In case the user wants to do something
47 // before generating primaries (i.e., store random number status), do it in
48 // the G4VUserPrimaryGeneratorAction concrete class.
49 //
50
51 class G4UserEventAction
52 {
53 public:
54     G4UserEventAction();
55     virtual ~G4UserEventAction();
56     inline void SetEventManager(G4EventManager* value)
57     { fpEventManager = value; }
58 public: // with description
59     virtual void BeginOfEventAction(const G4Event* anEvent);
60     virtual void EndOfEventAction(const G4Event* anEvent);
61     // Two virtual method the user can override.
62 protected:
63     G4EventManager* fpEventManager;
64 };
65
66 #endif
```

examples/extended/analysis/shared

```
37 #ifndef EventAction_h
38 #define EventAction_h 1
39
40 #include "G4UserEventAction.hh"
41 #include "globals.hh"
42
43 class RunAction;
44 class HistoManager;
45
46 //.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....ooo0000ooo.....
47
48 class EventAction : public G4UserEventAction
49 {
50 public:
51     EventAction(RunAction*, HistoManager*);
52     virtual ~EventAction();
53
54     virtual void BeginOfEventAction(const G4Event*);
55     virtual void EndOfEventAction(const G4Event*);
56
57     void AddAbs(G4double de, G4double dl) {fEnergyAbs += de; fTrackLABs += dl;};
58     void AddGap(G4double de, G4double dl) {fEnergyGap += de; fTrackLGap += dl;};
59
60 private:
61     RunAction* fRunAct;
62     HistoManager* fHistoManager;
63
64     G4double fEnergyAbs, fEnergyGap;
65     G4double fTrackLABs, fTrackLGap;
66
67     G4int fPrintModulo;
68 };
69
```

Geant4'te İz

- İz parçacığın “şipşak” ıdır (snapshot).
 - ▶ Sadece güncel durumun fiziksel niceliklerine sahiptir. Bir önceki nicelikleri barındırmaz.
 - ▶ Adım bir iz için “delta” bilgisidir. İz adımların bir toplamı değildir. Ancak iz adımlar tarafından güncellenir.
- İz nesnesi şu durumlarda silinir:
 - ▶ dünya hacminin dışına çıktığında,
 - ▶ ortadan yok olduğunda (bozunma, esnek olmayan çarpışma, soğurulma, ...),
 - ▶ sıfır kinetik enerjiye gittiğinde ilaveten “AtRest” bilgisi istenmediğinde
 - ▶ kullanıcı öldürmek istediğinde
- Hiç bir iz nesnesi olay sonunda yaşamaz.
 - ▶ İzlerin kaydı için gezinge sınıfının nesnelere kullanılmalıdır.
- **G4TrackingManager** sınıfı bir izin işlenmesinden sorumludur, **G4Track** sınıfı bir izi temsil eder.
- **G4UserTrackingAction** sınıfı ile kullanıcılar bu sınıfa ulaşıp işlem yapabilirler.

G4Track

```
32 // G4Track.hh
33 //
34 // Class Description:
35 // This class represents the particle under tracking.
36 // It includes information related to tracking for examples:
37 // 1) current position/time of the particle,
38 // 2) static particle information,
39 // 3) the pointer to the physical volume where currently
40 // the particle exists
41 //
42 //-----
43 // Modification for G4TouchableHandle          22 Oct. 2001 R.Chytracsek
44 // Add MaterialCutCouple                      08 Oct. 2002 H.Kurashige
45 // Add SetVelocityTableProperties             02 Apr. 2011 H.Kurashige
46 // Add fVelocity and Set/GetVelocity         29 Apr. 2011 H.Kurashige
47 // Use G4VelocityTable                        17 Aug. 2011 H.Kurashige
48
49 #ifndef G4Track_h
50 #define G4Track_h 1
51
52 #include <cmath> // Include from 'system'
53
54 #include "globals.hh" // Include from 'global'
55 #include "trkdefs.hh" // Include DLL defs...
56 #include "G4ThreeVector.hh" // Include from 'geometry'
57 #include "G4LogicalVolume.hh" // Include from 'geometry'
58 #include "G4VPhysicalVolume.hh" // Include from 'geometry'
59 #include "G4Allocator.hh" // Include from 'particle+matter'
60 #include "G4DynamicParticle.hh" // Include from 'particle+matter'
61 #include "G4TrackStatus.hh" // Include from 'tracking'
62 #include "G4TouchableHandle.hh" // Include from 'geometry'
63 #include "G4VUserTrackInformation.hh"
64 #include "G4PhysicsModelCatalog.hh"
65
66 #include "G4Material.hh"
67
68 class G4Step; // Forward declaration
69 class G4MaterialCutsCouple;
70 class G4VelocityTable;
71
72 ///////////////
73 class G4Track
74 ///////////////
75 {
76 //-----
77 public: // With description
78
79 // Constructor
80 G4Track();
81 G4Track(G4DynamicParticle* apValueDynamicParticle,
82         G4double aValueTime,
83         const G4ThreeVector& aValuePosition);
84 // aValueTime is a global time
85 G4Track(const G4Track&);
86 // Copy Constructor copies members other than tracking information
87
108 public: // With description
109 // Copy information of the track (w/o tracking information)
110 void CopyTrackInfo(const G4Track&);
111
112 // Get/Set functions
113 // track ID
114 G4int GetTrackID() const;
115 void SetTrackID(const G4int aValue);
116
117 G4int GetParentID() const;
118 void SetParentID(const G4int aValue);
119
120 // dynamic particle
121 const G4DynamicParticle* GetDynamicParticle() const;
122
123 // particle definition
124 const G4ParticleDefinition* GetParticleDefinition() const;
125 // following method of GetDefinition remains
126 // because of backward compatibility. It will be removed in future
127 G4ParticleDefinition* GetDefinition() const;
128
129 // position, time
130 const G4ThreeVector& GetPosition() const;
131 void SetPosition(const G4ThreeVector& aValue);
132
133 G4double GetGlobalTime() const;
134 void SetGlobalTime(const G4double aValue);
135 // Time since the event in which the track belongs is created.
136
137 G4double GetLocalTime() const;
138 void SetLocalTime(const G4double aValue);
139 // Time since the current track is created.
140
141 G4double GetProperTime() const;
142 void SetProperTime(const G4double aValue);
143 // Proper time of the current track
144
145 // volume, material, touchable
146 G4VPhysicalVolume* GetVolume() const;
147 G4VPhysicalVolume* GetNextVolume() const;
148
149 G4Material* GetMaterial() const;
150 G4Material* GetNextMaterial() const;
151
---
```

G4UserTrackingAction

G4UserTrackingAction.hh

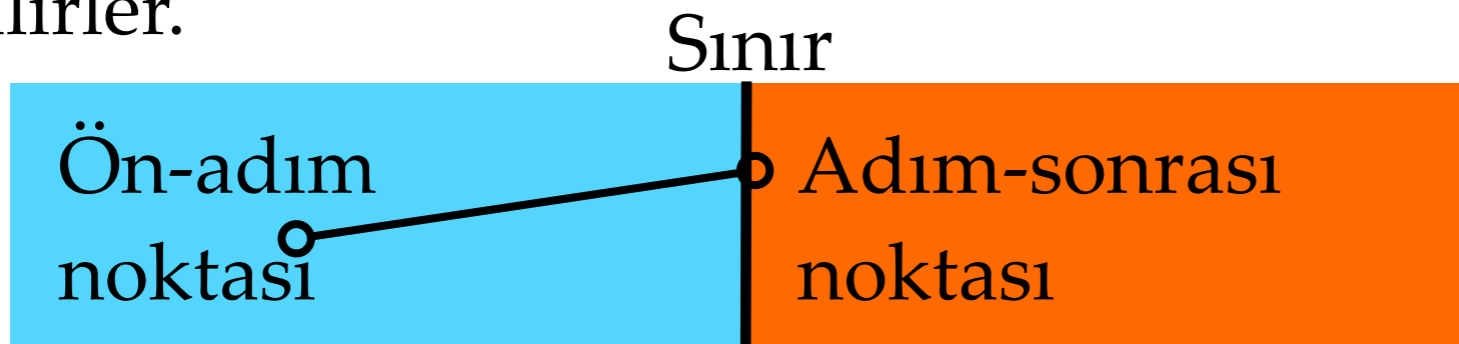
```
45 class G4UserTrackingAction;
46
47 #ifndef G4UserTrackingAction_h
48 #define G4UserTrackingAction_h 1
49
50 class G4TrackingManager;           // Forward declaration
51 class G4Track;
52
53 ///////////////////////////////////////////////////
54 class G4UserTrackingAction
55 ///////////////////////////////////////////////////
56 {
57
58 //-----
59 public: // with description
60 //-----
61
62 // Constructor & Destructor
63   G4UserTrackingAction();
64   virtual ~G4UserTrackingAction();
65
66 // Member functions
67   void SetTrackingManagerPointer(G4TrackingManager* pValue);
68   virtual void PreUserTrackingAction(const G4Track*);
69   virtual void PostUserTrackingAction(const G4Track*);
70
71 //-----
72   protected:
73 //-----
74
75 // Member data
76   G4TrackingManager* fpTrackingManager;
77
78 };
79
80 #endif
```

/examples/extended/hadronic/Hadr04/

```
34 #ifndef TrackingAction_h
35 #define TrackingAction_h 1
36
37 #include "G4UserTrackingAction.hh"
38 #include "globals.hh"
39
40 //....ooo0000ooo.....ooo0000ooo.....ooo0000ooo...
41
42 class TrackingAction : public G4UserTrackingAction {
43
44   public:
45     TrackingAction();
46     ~TrackingAction() {};
47
48     virtual void PreUserTrackingAction(const G4Track*);
49     virtual void PostUserTrackingAction(const G4Track*);
50
51     void UpdateTrackInfo(G4double, G4double, G4double);
52
53   private:
54     G4int fNbStep1, fNbStep2;
55     G4double fTrackLen1, fTrackLen2;
56     G4double fTime1, fTime2;
57 };
58
59 //....ooo0000ooo.....ooo0000ooo.....ooo0000ooo...
60
```

Geant4'te Adım

- Adım (step) iki noktaya sahiptir:
 - ▶ Ön-adım (pre-step) ve adım-sonrası (post-step)
- Adım parçacığın bir "delta" bilgisine de sahiptir:
 - ▶ adımdaki enerji kaybı, adım için geçen uçuş süresi,...
- Her nokta hacim ve malzeme bilgisine sahiptir. Adımı bir hacmin sınırlaması durumunda son nokta fiziksel olarak sınır üzerinde durur ve mantıksal olarak bir sonraki hacime ait olur.
 - ▶ Adımın iki hacmin malzemelerini bilmesinden dolayı geçiş ışıması ya da kırılma gibi süreçlerin benzetimi mümkün olmaktadır.
- **G4SteppingManager** sınıfı adımın işlenmesinden sorumludur, adım ise **G4Step** sınıfı ile temsil edilir.
- **G4UserSteppingAction** sınıfı ile kullanıcılar adım üzerinde işlemler yapabilirler.



G4Step

```
32 // G4Step.hh
33 //
34 // Class Description:
35 //   This class represents the Step of a particle tracked.
36 //   It includes information of
37 //     1) List of Step points which compose the Step,
38 //     2) static information of particle which generated the
39 //     Step,
40 //     3) trackID and parent particle ID of the Step,
41 //     4) termination condition of the Step,
42 //
43 // Contact:
44 //   Questions and comments to this code should be sent to
45 //     Katsuya Amako (e-mail: Katsuya.Amako@kek.jp)
46 //     Takashi Sasaki (e-mail: Takashi.Sasaki@kek.jp)
47 //
48 //
49 //
50 //
51 //
52 //
53 //
54 //
55 //
56 //
57 //
58 //
59 #ifndef G4Step_h
60 #define G4Step_h 1
61
62 #include <stdlib.h> // Include from 'system'
63 #include <cmath> // Include from 'system'
64 #include "G4ios.hh" // Include from 'system'
65 #include <iomanip> // Include from 'system'
66 #include "globals.hh" // Include from 'global'
67 #include "G4ThreeVector.hh" // Include from 'global'
68 #include "G4VPhysicalVolume.hh" // Include from 'geomet'
69 #include "G4StepPoint.hh" // Include from 'track'
70 #include "G4StepStatus.hh" // Include from 'track'
71 class G4Polyline; // Forward declaration.
72 class G4Track; // Forward declaration.
73 #include "G4TrackVector.hh" // Include from 'track'
74
75 ///////////////
76 class G4Step
77 ///////////////
78 {
79
80 //-----
81 public:
82
83 // Constructor/Destructor
84 G4Step();
85 ~G4Step();
86
87 // Copy Constructor and assignment operator
88 G4Step(const G4Step& );
89 G4Step & operator=(const G4Step &);
90
91 public: // With description
92
93
94 // Get/Set functions
95 // currnet track
96 G4Track* GetTrack() const;
97 void SetTrack(G4Track* value);
98
99 // step points
100 G4StepPoint* GetPreStepPoint() const;
101 void SetPreStepPoint(G4StepPoint* value);
102
103 G4StepPoint* GetPostStepPoint() const;
104 void SetPostStepPoint(G4StepPoint* value);
105
106 // step length
107 G4double GetStepLength() const;
108 void SetStepLength(G4double value);
109 // Before the end of the AlongStepDoIt loop, StepLength keeps
110 // the initial value which is determined by the shortest geometrical Step
111 // proposed by a physics process. After finishing the AlongStepDoIt,
112 // it will be set equal to 'StepLength' in G4Step.
113
114 // total energy deposit
115 G4double GetTotalEnergyDeposit() const;
116 void SetTotalEnergyDeposit(G4double value);
117
118 // total non-ionizing energy deposit
119 G4double GetNonIonizingEnergyDeposit() const;
120 void SetNonIonizingEnergyDeposit(G4double value);
121
```

G4UserSteppingAction

G4UserSteppingAction.hh

```
29 //-----
30 //
31 // G4UserSteppingAction.hh
32 //
33 // class description:
34 //   This class represents actions taken place by the user at each
35 //   end of stepping.
36 //
37 // Contact:
38 //   Questions and comments to this code should be sent to
39 //   Katsuya Amako (e-mail: Katsuya.Amako@kek.jp)
40 //   Takashi Sasaki (e-mail: Takashi.Sasaki@kek.jp)
41 //
42 //-----
43
44 #ifndef G4UserSteppingAction_h
45 #define G4UserSteppingAction_h 1
46
47 class G4Step;
48 class G4SteppingManager;           // Forward declaration
49
50 ///////////////////////////////////////////////////////////////////
51 class G4UserSteppingAction
52 ///////////////////////////////////////////////////////////////////
53 {
54 //-----
55 public: // with description
56 //-----
57
58 // Constructor and destructors
59 G4UserSteppingAction();
60 virtual ~G4UserSteppingAction();
61
62 // Member functions
63 void SetSteppingManagerPointer(G4SteppingManager* pValue);
64 virtual void UserSteppingAction(const G4Step*);
65
66 //-----
67 protected:
68 //-----
69
70 // Member data
71 G4SteppingManager* fpSteppingManager;
72
73 };
74
75 #endif
76
```

/examples/extended/hadronic/Hadr03

```
34 #ifndef SteppingAction_h
35 #define SteppingAction_h 1
36
37 #include "G4UserSteppingAction.hh"
38 #include "globals.hh"
39 #include <map>
40
41 class G4ParticleDefinition;
42 class RunAction;
43
44 //.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....
45
46 class SteppingAction : public G4UserSteppingAction
47 {
48 public:
49   SteppingAction(RunAction*);
50   ~SteppingAction();
51
52   virtual void UserSteppingAction(const G4Step*);
53
54 private:
55   RunAction* fRunAction;
56   std::map<G4ParticleDefinition*,G4int> fParticleFlag;
57 };
58
59 //.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....
60
```

/examples/extended/hadronic/Hadr03

```
55 void SteppingAction::UserSteppingAction(const G4Step* aStep)
56 {
57     // count processes
58     //
59     const G4StepPoint* endPoint = aStep->GetPostStepPoint();
60     const G4VProcess* process = endPoint->GetProcessDefinedStep();
61     fRunAction->CountProcesses(process);
62
63     // check that an real interaction occurred (eg. not a transportation)
64     G4StepStatus stepStatus = endPoint->GetStepStatus();
65     G4bool transmit = (stepStatus==fGeomBoundary || stepStatus==fWorldBoundary);
66     if (transmit) return;
67
68     //real processes : sum track length
69     //
70     G4double stepLength = aStep->GetStepLength();
71     fRunAction->SumTrack(stepLength);
72
73     //energy-momentum balance initialisation
74     //
75     const G4StepPoint* prePoint = aStep->GetPreStepPoint();
76     G4double Q = - prePoint->GetKineticEnergy();
77     G4ThreeVector Pbalance = - prePoint->GetMomentum();
78
79     //initialisation of the nuclear channel identification
80     //
81     G4ParticleDefinition* particle = aStep->GetTrack()->GetDefinition();
82     G4String partName = particle->GetParticleName();
83     G4String nuclearChannel = partName;
84     G4HadronicProcess* hproc = (G4HadronicProcess*) process;
85     const G4Isotope* target = hproc->GetTargetIsotope();
86     G4String targetName = "XXXX";
87     if (target) targetName = target->GetName();
88     nuclearChannel += " + " + targetName + " --> ";
89
90     //scattered primary particle (if any)
91     //
92     G4AnalysisManager* analysis = G4AnalysisManager::Instance();
93     G4int ih = 1;
94     if (aStep->GetTrack()->GetTrackStatus() == fAlive) {
95         G4double energy = endPoint->GetKineticEnergy();
96         analysis->FillH1(ih,energy);
97         //
98         G4ThreeVector momentum = endPoint->GetMomentum();
99         Q += energy;
100        Pbalance += momentum;
101        //
102        nuclearChannel += partName + " + ";
103    }
104
105     //secondaries
106     //
107     const std::vector<const G4Track*>* secondary
108         = aStep->GetSecondaryInCurrentStep();
109     for (size_t lp=0; lp<(*secondary).size(); lp++) {
110         particle = (*secondary)[lp]->GetDefinition();
111         G4String name = particle->GetParticleName();
112         G4String type = particle->GetParticleType();
113         G4double charge = particle->GetPDGCharge();
114         G4double energy = (*secondary)[lp]->GetKineticEnergy();
115         fRunAction->ParticleCount(name,energy);
116         //energy spectrum
117         if (charge > 3.) ih = 2;
118         else if (particle == G4Gamma::Gamma()) ih = 3;
119         else if (particle == G4Neutron::Neutron()) ih = 4;
120         else if (particle == G4Proton::Proton()) ih = 5;
121         else if (particle == G4Deuteron::Deuteron()) ih = 6;
122         else if (particle == G4Alpha::Alpha()) ih = 7;
123         else if (type == "nucleus") ih = 8;
124         else if (type == "meson") ih = 9;
125         else if (type == "baryon") ih = 10;
126         analysis->FillH1(ih,energy);
127         //energy-momentum balance
128         G4ThreeVector momentum = (*secondary)[lp]->GetMomentum();
129         Q += energy;
130         Pbalance += momentum;
131         //particle flag
132         fParticleFlag[particle]++;
133     }
134
135     //energy-momentum balance
136     G4double Pbal = Pbalance.mag();
137     fRunAction->Balance(Pbal);
138     ih = 11;
139     analysis->FillH1(ih,Q);
140     ih = 12;
141     analysis->FillH1(ih,Pbal);
142 }
```


Gezinge ve Gezinge Noktası

- İz kendisini takip etmez. İz nesnelere olay sonunda yok olur.
- **G4Trajectory** sınıfı G4Track sınıfının **bazı** bilgilerini kopyalar.
- **G4TrajectoryPoint** sınıfı G4Step sınıfının **bazı** bilgilerini kopyalar.
 - ▶ G4Trajectory sınıfı G4TrajectoryPoint sınıfının bir vektörüne sahiptir.
 - ▶ Olay işleminin sonunda, G4Event sınıfı G4Trajectory nesnelere bir topluluğuna sahiptir (bunun için ise)
 - /tracking/storeTrajectory must be set to 1.
- Aradaki farka dikkat edilmelidir!
 - ▶ G4Track \Leftrightarrow G4Trajectory, G4Step \Leftrightarrow G4TrajectoryPoint
- G4Trajectory ve G4TrajectoryPoint nesnelere olay sonuna kadar var olduklarından öök fazla izi kaydetmemeye dikkat edilmelidir.
 - ▶ örneğın yüksek enerjili EM sađanak izlerini saklamaktan kaçınılmalıdır.
- G4Trajectory ve G4TrajectoryPoint sadece minimum bilgi içerirler.
 - ▶ Kullanıcılar bilgi tutmak için kendi gezinge ve gezinge noktası sınıflarını G4VTrajectory ve G4VTrajectoryPoint baz sınıflarından oluşturabilirler.

Geant4'te Parçacık

- Geant4'te bir parçacık üç sınıf ile temsil edilir: **G4Track**, **G4DynamicParticle**, **G4ParticleDefinition**.
- **G4Track**
 - ▶ Konum, geometrik bilgi, ...
 - ▶ Bu sınıf izlenecek olan parçacığı temsil eder.
- **G4DynamicParticle**
 - ▶ Parçacığın momentum, enerji gibi "dinamik" fiziksel özelliklerini temsil eder.
 - ▶ Her bir G4Track nesnesi kendisine ait ve biricik (unique) bir G4DynamicParticle nesnesine sahiptir.
 - ▶ Bu sınıf özgün bir parçacığı temsil eder.
- **G4ParticleDefinition**
 - ▶ Parçacığın spin, yük, kütlesi gibi "statik" özelliklerini temsil eder. G4ProcessManager parçacığın yer aldığı süreci tanımlayan sınıftır.
 - ▶ Aynı bir parçacığın tüm G4DynamicParticle nesneleri aynı G4ParticleDefinition sınıfını paylaşırlar.

İzleme ve Süreçler

Geant4 izlemesi geneldir. Yani

- Parçacık cinsinden
- Parçacığı içeren fizik süreçlerinden

bağımsızdır.

Tanımlı olan tüm süreçlere şans (gerçekleşme olasılığı) verir, böylece her süreç

- adım uzunluğunu belirlemek için
- izin fiziksel özelliklerinde olası değişiklik yapmak için
- ikincil parçacıkları yaratmak için
- izin durumunda bazı değişiklikler önermek için (beklet, durdur, ötele, öldür)

katkıda bulunabilir.

Geant4'te Süreçler

- Geant4'te parçacık taşınması da bir süreçtir. Bu süreçte parçacık geometrik hacmin sınırları ile ve herhangi bir alan ile etkileşir.
- Her parçacık kendisine uygulanabilir süreçlerin bir listesine sahiptir. Her adımda, listelenen süreçler önerilen fiziksel etkileşme uzunluğunu bulmak için çağırılır.
- (Uzay-zamanda) en kısa etkileşim uzunluğunu veren süreç adımı sınırlandırır.
- Her süreç aşağıdaki durumların birisine ya da bunların karışımına sahiptir:
 - ▶ AtRest
 - örneğin durgun durumdaki muon'un bozunması
 - ▶ AlongStep (b.d.a. sürekli süreç)
 - örneğin Cerenkov süreci
 - ▶ PostStep (b.d.a. kesikli süreç)
 - örneğin uçuş sırasında bozunma

Kullanışlı Bilginin Elde Edilmesi

- Geometri, fizik ve birincil iz üretiminin verilmesi ile Geant4 “sessizce” fizik simülasyonu yapar.
 - ▶ Sizin için kullanışlı bilgiyi elde etmek için biraz (!) kod yazmanız gerekir.
- Bunun iki yolu vardır:
 - ▶ Baz sınıflara verilmiş olan çengelleri kullanabilirsiniz (G4UserTrackingAction, G4UserSteppingAction, ...)
 - Neredeyse tüm bilgiye ulaşabilirsiniz
 - Dolambaçsızdır, ancak kendiniz yapmalısınız
 - ▶ Geant4’ün sayım (scoring) işlevselliğini kullanabilirsiniz
 - Bir hacime G4VSensitiveDetector atayın
 - Vuru topluluğu G4Event’te otomatik olarak saklanır ve eğer kullanıcı tarafından tanımlanmış bir Run nesnesi kullanılmış ise otomatik olarak biriktirilir.
 - Olay ve koşu özetini almak için yine çengelleri kullanabilirsiniz (G4UserEventAction, G4UserRunAction)

Birim Sistemi

- Geant4'te kullanılan iç birim sistemi hem kullanıcıdan hem de Geant4'ün kaynak kodundan tamamen gizlenmiştir.
- Doğrudan kodlanmış (hard-coded) her sayı uygun birimi ile çarpılmalıdır.
radius = 10.0 * **cm**;
kineticE = 1.0 * **GeV**;
- Sayı elde etmek için uygun birimi ile bölünmelidir.
G4cout << eDep / **MeV** << " [MeV]" << G4endl;
- Sıklıkla kullanılan birimlerin çoğu tanımlanmıştır ve kullanıcı kendi birimini ekleyebilir.
- Bu birim sistemi ile kaynak kodu daha okunaklı hale gelir ve fiziksel nicelikleri koda ekleyip çıkartmak daha basit olur.
 - ▶ Özel bir uygulama için kullanıcı iç birimi, sonucu değiştirmeden, daha uygun bir alternatif birime çevirebilir.

G4cout, G4cerr

- G4cout ve G4cerr (G4endl ile birlikte) Geant4 tarafından tanımlanmış ostream nesnelere aittir.
G4cout << "Hello Geant4!" << G4endl;
- Bazı GUI'ler çıktı karakter dizilerini arabelleğe yüklerler ve çıktıları başka pencerelerde gösterirler ya da saklama/değiştirme işlevselliği sunarlar
 - ▶ Kullanıcı std::cout, std::cerr vb. kullanmamalıdır.
- Girdi için std::cin kullanılmamalıdır. Sağlanmış olan komutlar kullanılmalıdır
 - ▶ Bir dosya için bilindik I/O yapabilirsiniz.

Geant4 Kullanımı

- Geant4 bir parçalardan oluşmuş bir yazılımdır (toolkit). Bu parçalardan bir uygulama inşa etmelisiniz.
- Bir uygulama oluşturmak için
 - ▶ Geometrik düzeni tanımlayın
 - Malzeme, hacim
 - ▶ Gerekli olan fiziği tanımlayın
 - Parçacıklar, fizik süreçleri/modeller
 - Üretim eşikleri
 - ▶ Bir olayın nasıl başladığını tanımlayın
 - Birincil izin üretimi
 - ▶ İşinize yarayacak bilgileri çıkartın
- Bunların yanı sıra, isterseniz
 - ▶ Geometriyi, izleri ve fizik çıktılarını görselleştirin
 - ▶ Kullanıcı arayüzeyini kullanın G(UI)
 - ▶ Kendi UI komutlarınızı tanımlayın
 - ▶ ...

Kullanıcı Sınıfları

- **main()**
 - ▶ Geant4 main()'i sağlamaz, kullanıcı bu programı yazmalıdır.
- Başlatma durumuna getirme sınıfları
 - ▶ Tanımlamak için G4RunManager::**SetUserInitialization()** kullanın.
 - ▶ Başlatma durumuna getirilmede çağırılanlar
 - **G4VUserDetectorConstruction**
 - **G4VUserPhysicsList**
- Eylem sınıfları
 - ▶ Tanımlamak için G4RunManager::**SetUserAction()** kullanın
 - ▶ Bir olay çeviriminde çağrılanlar
 - **G4VUserPrimaryGeneratorAction**
 - G4UserRunAction
 - G4UserEventAction
 - G4UserStackingAction
 - G4UserTrackingAction
 - G4UserSteppingAction

Not: **Kırmızı** ile yazılı olan sınıflar zorunlu olarak kullanılmalıdır.

main() Programı

- Geant4 main()'i sağlamaz, kullanıcı bu programı yazmalıdır.
- main() programında
 - ▶ G4RunManager (ya da bundan türetmiş olduğunuz sınıfı) oluşturmalısınız
 - ▶ Kullanıcının **yazması zorunlu olan sınıfları** RunManager'a ayarlamalısınız:
 - G4VUserDetectorConstruction
 - G4VUserPhysicsList
 - G4VUserPrimaryGeneratorAction
- main() programınızda VisManager, (G)UI oturumunu, tercih ettiğiniz sizin tanımlamış olduğunuz eylem sınıflarını tanımlayabilirsiniz.

Algıç Tanımı

- Kendi somut (concrete) sınıfınızı **G4VUserDetectorConstruction** soyut baz sınıfından türetin.
- Sanal Construct() metodunda,
 - ▶ Tüm gerekli malzemeleri örnekleyin
 - ▶ Algıç geometrisini oluşturan hacimleri örnekleyin
 - ▶ hassas algıç sınıflarınızı örnekleyin ve bunları karşılık geldikleri mantıksal hacimlere atayın
- Tercihen
 - ▶ algıcınızın herhangi bir parçası için bölgeler (Regions)
 - ▶ algıç elemanları için görüntüleme özellikleri (renk, görünürlük vb.) tanımlayabilirsiniz.

Fizik Süreçlerinin Seçimi

Geant4'te varsayılan parçacıklar veya süreçler yoktur.

- ▶ Parçacık taşınması için bile bunu açıkça tanımlamak gerekir.

Fizik süreçlerini içeren somut sınıfı soyut **G4VUserPhysicsList** baz sınıfından türetilir.

- ▶ Gerekli tüm parçacıkları tanımlayın.
- ▶ Gerekli tüm süreçleri tanımlayın ve bunları süreçlerin uygulanabileceği parçacıklara atayın.
- ▶ Hem dünya geometrisine hem de tüm bölgelere uygulanacak olan kesme değer aralıklarını tanımlayın

Geant4 pek çok işe yarar sınıf/metod ve örnek sağlamaktadır.

- ▶ Hadronik süreçler için, pek çok farklı kullanma durumu için, "deneyime dayanan tahmini" fizik listeleri vardır.

Birincil Olayın Üretilmesi

- Somut sınıfınızı soyut **G4VUserPrimaryGeneratorAction** baz sınıfından türetin.
- Bir G4Event nesnesini bir ya da birden fazla birincil köşe ya da birincil parçacık üreten somut birinci üretme sınıfına gönderin.
- Geant4, G4VPrimaryParticleGenerator baz sınıfının yanı sıra bazı üreticileri de sağlar:
 - ▶ G4ParticleGun
 - ▶ G4HEPEvtInterface, G4HepMCInterface
 - ▶ HepMC sınıfı yada /hepevt/ ortak bloğu için arayüzey
 - ▶ G4GeneralParticleSource
 - ▶ Radyoaktiviteyi tanımlar

İsteğe Bağlı Kullanıcı Eylem Sınıfları

- All user action classes, methods of which are invoked during “Beam On”, must be constructed in the user’s *main()* and must be set to the RunManager.
- **G4UserRunAction**
 - ▶ G4Run* GenerateRun()
 - Kullanıcının özelleştirdiği koşu nesnesini örnekleme
 - ▶ void BeginOfRunAction(const G4Run*)
 - Histogram tanımlama
 - ▶ void EndOfRunAction(const G4Run*)
 - Koşuyu inceleme
 - Histogramları depolama
- **G4UserEventAction**
 - ▶ void BeginOfEventAction(const G4Event*)
 - Olayı seçimi
 - ▶ void EndOfEventAction(const G4Event*)
 - Olay çıktı bilgisi oluşturma

Kullanıcı Dökümanları

<http://geant4.cern.ch/support/userdocuments.shtml>

User's Guides

The following four documents comprise a complete set of user's manuals for the Geant4 toolkit. The

Installation Guide: For setting up Geant4 in your computing environment [[pdf version](#)]

We strongly recommend installing the Geant4 toolkit under your computing environment before you are reading them. This installation guide instructs you in the setting up of

[Old Versions]

- Version 9.6: [html](#), [pdf](#)
- Version 9.5: [html](#), [pdf](#)

User's Guide: For Application Developers [[pdf version](#)]

If you are new to Geant4, we recommend that you read this document first. The first part of the toolkit for practical applications, with a lot of example codes. After reading this part, you will be able to use the advanced use of the toolkit.

[Old Versions]

- Version 9.6: [html](#), [pdf](#)
- Version 9.5: [html](#), [pdf](#)

User's Guide: For Toolkit Developers [[pdf version](#)]

This document is for those who want to contribute to the extension of the functionality to the toolkit. It contains the analysis and design performed by the original toolkit developers. Understanding this design is essential for the development of new features.

[Old Versions]

- Version 9.6: [html](#), [pdf](#)
- Version 9.5: [html](#), [pdf](#)

Physics Reference Manual [[Note](#)]

This is a detailed description of the physics interactions provided in the Geant4 toolkit. The Physics Reference Manual contains gaps in documentation which correspond to un-implemented features in the next release.

[Old Versions]

- Version 9.6: [pdf](#)
- Version 9.5: [pdf](#)

Browsing Geant4 source codes

For those who want to know details of the Geant4 codes,

- [Code Cross References - LXR](#)
- [Classes and Members Reference Guide - Doxygen](#)

Geant4 Cross Reference (LXR)

<http://www-geant4.kek.jp/LXR/>

Geant 4

Geant4 Cross Reference

Cross-Referencing Geant4

Version: [[ReleaseNotes](#)] [[1.0](#)] [[1.1](#)] [[2.0](#)] [[3.0](#)] [[3.1](#)] [[3.2](#)] [[4.0](#)] [[7.0](#)] [[7.0.p1](#)] [[7.1](#)] [[7.1.p1](#)] [[8.0](#)] [[8.0.p1](#)] [[8.1](#)] [[8.1.p1](#)] [[8.1.p2](#)] [[8.1.p3](#)] [[8.1.p4](#)] [[9.2.p4](#)] [[9.3](#)] [[9.3.p1](#)] [[9.3.p2](#)] [[9.4](#)] [[9.4.p1](#)] [[9.4.p2](#)] [[9.4.p3](#)] [[9.4.p4](#)]

Find file:

[/physics_lists/constructors/limiters/include/G4StepLimiterPhysics.hh](#)
[/physics_lists/constructors/limiters/src/G4StepLimiterPhysics.cc](#)
[/processes/transportation/include/G4StepLimiter.hh](#)
[/processes/transportation/src/G4StepLimiter.cc](#)
[/track/include/G4SteppingControl.hh](#)
[/track/include/G4Step.icc](#)
[/track/include/G4Step.hh](#)
[/track/include/G4StepStatus.hh](#)
[/track/include/G4StepPoint.icc](#)
[/track/include/G4StepPoint.hh](#)
[/track/src/G4StepPoint.cc](#)
[/track/src/G4Step.cc](#)
[/tracking/include/G4SteppingVerbose.hh](#)
[/tracking/include/G4SteppingManager.hh](#)
[/tracking/src/G4SteppingManager2.cc](#)
[/tracking/src/G4SteppingVerbose.cc](#)
[/tracking/src/G4SteppingManager.cc](#)
[/environments/g4py/source/track/pyG4Step.cc](#)
[/environments/g4py/source/track/pyG4StepStatus.cc](#)
[/environments/g4py/source/track/pyG4StepPoint.cc](#)

Geant4 Reference Guide (Doxygen)

<http://www-geant4.kek.jp/Reference/>

Geant4 9.6.p02

Main Page Namespaces Classes Files

Geant4

- Namespaces
- Classes
- Files

Geant4 Documentation

Geant4 9.6.p02

Main Page Namespaces Classes Files

Class List Class Index Class Hierarchy Class Members

- G4StatMF
- G4StatMFChannel
- G4StatMFFragment
- G4StatMFMacroBiNucleon
- G4StatMFMacroCanonical
- G4StatMFMacroChemicalPotential
- G4StatMFMacroMultiNucleon
- G4StatMFMacroMultiplicity
- G4StatMFMacroNucleon
- G4StatMFMacroTemperature
- G4StatMFMacroTetraNucleon
- G4StatMFMacroTriNucleon
- G4StatMFMicroCanonical
- G4StatMFMicroManager

G4Step Class Reference

Public Member Functions

- G4Step ()
- ~G4Step ()
- G4Step (const G4Step &)
- G4Step & operator= (const G4Step &)
- G4Track * GetTrack () const
- void SetTrack (G4Track *value)
- G4StepPoint * GetPreStepPoint () const
- void SetPreStepPoint (G4StepPoint *value)

İsteğe Bağlı Kullanıcı Eylem Sınıfları

- **G4UserStackingAction**

- ▶ void PrepareNewEvent()

- Öncelik kontrolünü yeniden ayarar

- ▶ G4ClassificationOfNewTrack ClassifyNewTrack(const G4Track*)

- Yeni bir iz yığına itildiği her zaman çağrılır
- Yeni bir izi sınıflandırır -- önem kontrolü

- ➔ Urgent, Waiting, PostponeToNextEvent, Kill

- ▶ void NewStage()

- Urgent yığını boşaldığında çağrılır
- Sınıflandırma kriterini değiştirir
- Olay filtrelemesi (Olay durdurması)

İsteğe Bağlı Kullanıcı Eylem Sınıfları

- **G4UserTrackingAction**

- ▶ void PreUserTrackingAction(const G4Track*)
 - Gezingenin saklanıp saklanmayacağına karar verin
 - Kullanıcı tarafından tanımlanmış gezingeyi yaratın
- ▶ void PostUserTrackingAction(const G4Track*)
 - Gereksiz olan gezinmeleri silin

- **G4UserSteppingAction**

- ▶ void UserSteppingAction(const G4Step*)
 - İzi Öldür / Beklet / Ertele (kill/suspend/postpone)
 - Adımı çizin (gezinge olarak saklanmayacak izler için)

Özetle:

Malzeme ve geometriyi tanımlayın

➔ G4VUserDetectorConstruction

Uygun parçacık ve süreçleri seçin, ve üretim eşiklerini tanımlayın

➔ G4VUserPhysicsList

Birincil parçacığın üretim biçimini tanımlayın

➔ G4VUserPrimaryGeneratorAction

Geant4'ten işe yarar bilgiyi nasıl alacağınızı tanımlayın

➔ G4UserSteppingAction, G4UserTrackingAction, vb.

➔ G4VUserDetectorConstruction, G4UserEventAction, G4Run, G4UserRunAction

➔ G4SensitiveDetector, G4VHit, G4VHitsCollection

Kurulum

Desteklenen Platformlar Ve Derleyiciler

<http://geant4.web.cern.ch/geant4/support/ReleaseNotes4.10.0.html>

- Linux

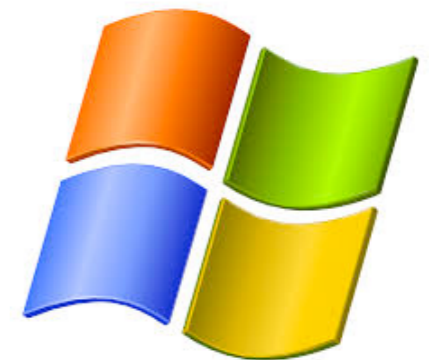
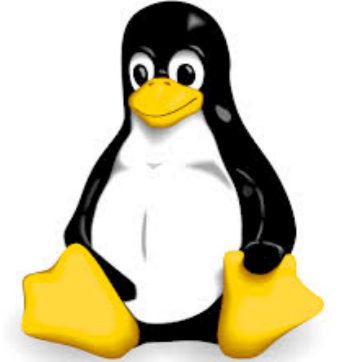
- Resmi olarak test edilen: 64 bit (Intel ya da AMD) Scientific Linux CERN SLC6, gcc-4.4.7.
- Sadece teyit edilen: gcc-4.7.2, gcc-4.8.2 (64 bit)

- Mac OSX

- Resmi olarak test edilen: 0.8, gcc-4.2.1 ve clang-3.3
- Sadece teyit edilen: 10.9 with gcc-4.2.1 and clang-3.3

- Windows

- Resmi olarak test edilen: Windows7, Visual C++ 11.0 (Visual Studio 2012)
- Sadece teyit edilen: Windows7, Visual C++ 9.0, 10.0, 12.0



Gerekli Yazılımlar

- Bir UNIX kabuğu ve gerekli unix komutları
- Geant4 kaynak kodu
- CMake, 2.6.4 ya da daha yenisi
 - ▶ Önerilen ise 2.8.X
- C++ Derleyicisi
 - ▶ Linux: GNU Derleyici Toplamı
 - ▶ Mac Os X: Clang (Xcode4)
 - Xcode4 için komut satırı araçları gereklidir (Xcode → Preferences → Downloads)
 - Xcode5 için eğer komut satırı araçları GUI'den yüklenemezse komut satırından `xcode-select --install` işlemini yapın.
- Windows: Visual Studio 2011 (Express ya da Higher Ed.)
- Geant4 veri dosyaları (eğer CMake otomatik olarak alamazsa)

Tercih edilen kurulum biçimi CMake ile dir.

<http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/InstallationGuide/html/ch01s02.html>

Dış Yazılımlar-1

Görüntüleme/GUI Araçları (tercihen):

- X11 OpenGL Visualization (Linux ve Mac OS X)
 - ▶ Kurulması Gerekir: X11, OpenGL or MesaGL (üstlük ve kütüphaneler).
- Qt4 User Interface and Visualization (Tüm Platformlar)
 - ▶ Kurulması Gerekir: Qt4, OpenGL or MesaGL (headers and libraries).
- Motif User Interface and Visualization (Linux ve Mac)
 - ▶ Kurulması Gerekir: Motif and X11, OpenGL or MesaGL headers and libraries.
- Open Inventor Visualization (Tüm Platformlar)
- X11 RayTracer Visualization (Linux ve Mac OS X)
- GDML Desteği (Tüm Platformlar)
- DAWN postscript yorumlayıcısı
- HepRApp Browser
- VRMLbrowser
- WIRED4 JAS Plug-In

kaynak: <https://agenda.infn.it/getFile.py/access?sessionId=5&resId=0&materialId=0&confId=5699>

Dış Yazılımlar-2

Analiz ve Histogramlar için Yazılımlar(tercihen):

- AIDA (Abstract Interfaces for Data Analysis)
 - ▶ iAIDA (an implementation of AIDA'nın C++ uygulanması)
 - ▶ JAS3 (Java Analysis Studio)
 - ▶ Open Scientist (Interactive Analysis Environment)
 - ▶ rAIDA (AIDA'nın Root'a uygulanması)

[http:// aida.freehep.org](http://aida.freehep.org)

- ROOT (veri analizi yazılımı)

<http://root.cern.ch>

Nelere İhtiyacınız Var:

Geant4 Software Download

Geant4 10.0

released 6 December 2013

The Geant4 source code is freely available. See the [licence conditions](#).

Please read the [Release Notes](#) before downloading or using this release. It is required to apply a full rebuild of the libraries.

Source files

Please choose the archive best suited to your system and archiving tool:

- [Download](#) GNU or Linux tar format, compressed using gzip (29.4Mb, 30780131 bytes)
After downloading, gunzip, then unpack using [GNU tar](#).
- [Download](#) ZIP format (41.4Mb, 43365939 bytes)
After downloading, unpack using e.g. WinZip.

Data files (*)

For specific, optional physics processes some of the following files are required. The file format is compatible with Unix, GNU, and Windows utilities.

- [Download](#) Neutron data files with thermal cross-sections - version 4.4 (402.0Mb, 421555304 bytes) **NEW**
- [Download](#) Data files for low energy electromagnetic processes - version 6.35 (18.2Mb, 19092577 bytes) **NEW**
- [Download](#) Data files for photon evaporation - version 3.0 (8.5Mb, 8864188 bytes) **NEW**
- [Download](#) Data files for radioactive decay hadronic processes - version 4.0 (962.4kb, 985509 bytes) **NEW**
- [Download](#) Data files from evaluated cross-sections in SAID data-base - version 1.1 (25.2kb, 25800 bytes)
- [Download](#) Data files for evaluated neutron cross-sections on natural composition of elements - version 1.4 (2.1Mb, 2249001 bytes) **NEW**
- [Download](#) Data files for nuclear shell effects in INCL/ABLA hadronic mode - version 3.0 (53.6kb, 54849 bytes) **NEW**
- [Download](#) Data files for shell ionisation cross-sections - version 1.3 (4.1Mb, 4293607 bytes)
- [Download](#) Optional data files for measured optical surface reflectance - version 1.0 (1.2Mb, 1257863 bytes)
- [Download](#) Optional data files for nuclides properties - version 1.0 (229.1kb, 234612 bytes) **NEW**

- Low Energy Nuclear Data (LEND) files can be downloaded from: <http://gdo-nuclear.ucllnl.org/pub/>

Pre-compiled Libraries

These are compiled with Geant4 default settings and optimization turned on. Please choose according to your system/compiler:

- [Download](#) compiled using gcc 4.4.7 on Scientific Linux CERN 6 (SLC6, based on Redhat Linux Enterprise 6), 64 bits (15.0Mb, 15684036 bytes)
- [Download](#) compiled using gcc 4.2.1/clang-3.3 on Mac (MacOSX 10.9), 64 bits (13.6Mb, 14253160 bytes)
- [Download](#) compiled using VC++ 11.0 on Windows 7, 32 bits, zip file (48.3Mb, 50631960 bytes)
- [Download](#) compiled using VC++ 11.0 on Windows 7, 32 bits, executable installer (34.9Mb, 36606241 bytes)

These libraries are built using the embedded CLHEP classes, based on [CLHEP version 2.1.4.1](#). Please refer to the [Release Notes](#) for details.

CMake ile Kurulum

- Geant4 kaynak paketini (geant4.10.00-2.tar.gz) istediğiniz bir yere açın:
 - ▶ örneğin: /path/to/geant4.10.00 (kaynak altdizini)
- Kurulum ürünlerini yapılandıracak, çalıştıracak ve kuracak yeni bir altdizin yaratın (kaynak altdizinininden farklı bir yer):
 - ▶ örneğin: /path/to/geant4.10.00-build (kurulum altdizini)

```
$ cd /path/to
$ mkdir geant4.10.00-build
$ ls
geant4.10.00 geant4.10.00-build
```

- Ayarları yapmak için kurulum altdizinine gidin ve CMake'i çalıştırın:

```
$ cd /path/to/geant4.10.00-build
$ cmake -DCMAKE_INSTALL_PREFIX= /path/to/geant4.10.00-install /path/to/geant4.10.00
```

Kurulum yapmayı tercih ettiğimiz altdizin

Geant4'ün kaynağının bulunduğu yer

CMake ile İleri Kurulum

Ek argümanlar CMake'e verilerek Geant4'ün tercihli olan paketleri de kurulabilir:

- **-DGEANT4_INSTALL_DATA=ON** (önerilir)

ek olan veri kütüphaneleri otomatik olarak indirilir

- **-DGEANT4_INSTALL_EXAMPLES=ON** (önerilir) örnekler kurulur

- **-DGEANT4_USE_OPENGL_X11=ON** (önerilir) X11 OpenGL

görüntüleme sürücüsünü kurar

- **-DGEANT4_USE_QT=ON** (tercihli) Qt görüntüleme sürücüsünü kurar

En başından bu seçenekler CMake'eklenebilir:

```
$ cmake -DCMAKE_INSTALL_PREFIX=/path/to/geant4.9.6-install -DGEANT4_INSTALL_DATA=ON \  
-DGEANT4_USE_OPENGL_X11=ON -DGEANT4_INSTALL_EXAMPLES=ON /path/to/geant4.9.6
```

CMake ile Kurulum

CMake kurulum için ayarlamaları yapar ve asıl kurumu başlatmak için Unix Makefile dosyalarını üretir:

```
$ cmake -DCMAKE_INSTALL_PREFIX=/path/to/geant4.10.00-install /path/to/geant4.10.00
-- The C compiler identification is GNU 4.2.1
-- The CXX compiler identification is GNU 4.2.1
-- Checking whether C compiler has -isysroot
-- Checking whether C compiler has -isysroot - yes
-- Checking whether C compiler supports OSX deployment target flag
-- Checking whether C compiler supports OSX deployment target flag - yes
-- Check for working C compiler: /Developer/usr/bin/cc
-- Check for working C compiler: /Developer/usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Checking whether CXX compiler has -isysroot
-- Checking whether CXX compiler has -isysroot - yes
-- Checking whether CXX compiler supports OSX deployment target flag
-- Checking whether CXX compiler supports OSX deployment target flag - yes
-- Check for working CXX compiler: /Developer/usr/bin/c++
-- Check for working CXX compiler: /Developer/usr/bin/c++ -- works
```

Herşey yolunda giderse:

```
-- Configuring done
-- Generating done
-- Build files have been written to: /path/to/geant4.10.00-build
```

Bu mesajları görmezseniz CMake'in hata mesajlarını dikkatlice kontrol edin.

CMake ile Kurulum

\$ make -jN (N: kaç tane paralel iş istiyoruz.)

\$ make install (MAKE_INSTALL_PREFIX altdizinine kurulum)

Geant4'ün kurulumu platformunuzun hızına bağlı olarak bir süre alacaktır.

Eğer sorun yaşadıysanız, çözüm daha önce önerilmiş olabilir:

<http://hypernews.slac.stanford.edu/HyperNews/geant4/get/installconfig.html>

GEANT4 at hypernews.slac.stanford.edu Forum List by Category Not Logged In ([login](#))

Geant 4 [Forums by Category](#) [Recent Postings](#) [Member Info](#) [Overview](#)
[Forums by Time Order](#) [Search in Forums](#) [Members List](#) [Contact Admin](#)
[Request a New Forum](#) [Subscribe to Forums](#) [New Member](#)

[Page Help](#)

Category: Applications

[Educational Applications](#) [Industrial instruments](#) [Medical Applications](#) [Space Applications](#)

Category: Control of runs, events, tracks, particles

[Event and Track Management](#) [Multithreading](#) [Particles](#) [Run Management](#)

Category: Experimental Setup

[Biasing and Scoring](#) [Fields: Magnetic and Otherwise](#) [Geometry](#) [Hits, Digitization and Pileup](#)

Category: General matters

[Documentation and Examples](#) [HyperNews System Announcements](#) [Hypernews Testing](#) [Installation and Configuration](#) [User Requirements](#)

Category: Interfaces

[\(Graphical\) User Interfaces](#) [Analysis](#) [Persistency](#) [Visualization](#)

Category: Physics

[Biasing and Scoring](#) [Electromagnetic Processes](#) [Fast Simulation, Transportation & Others](#) [Hadronic Processes](#) [Physics List](#) [Processes Involving Optical Photons](#)

Kurulum Bitince:

```
+-- CMAKE_INSTALL_PREFIX
+- bin/
| +- geant4-config (UNIX ONLY)
| +- geant4.csh (UNIX ONLY)
| +- geant4.sh (UNIX ONLY)
| +- G4global.dll (WINDOWS ONLY)
| +- ...
+- include/
| +- Geant4/
| | +- G4global.hh
| | +- ...
| | +- CLHEP/ (WITH INTERNAL CLHEP ONLY)
| | +- tools/
+- lib/ (MAY BE lib64 on LINUX)
| +- libG4global.so (AND/OR .a, OR G4Global.lib ON WINDOWS)
| +- ...
| +- Geant4-10.0.0/
| | +- Geant4Config.cmake
| | +- Geant4ConfigVersion.cmake
| | +- Geant4LibraryDepends.cmake
| | +- Geant4LibraryDepends-Release.cmake
| | +- UseGeant4.cmake
| | +- Linux-g++ (OR Darwin-g++ UNIX ONLY SOFTLINK -> ..)
| | +- Modules/
+- share
| +- Geant4-10.0.0
| | +- data/ (IF GEANT4_INSTALL_DATA WAS SET)
| | +- examples/
| | +- geant4make/
| | | +- geant4make.csh
| | | +- geant4make.sh
| | | +- config/
```

/path/to/geant4.10.00-install

Derlenmiş kütüphanelere PATH değişkeninden ulaşmak ve veri kütüphanelerine ulaşmak için:

\$. geant4.sh
ya da
\$. geant4.csh

CMake ile Bir Uygulamayı İnşa Etmek

Geant4'ün temel örneklerinden birisini inşa edelim:

geant4.10.00/examples/basic/B1/

```
$ cd $HOME
$ mkdir B1-build
$ cd B1-build
$ cmake -DGeant4_DIR=/path/to/geant4.10.00-install/lib(64)/Geant4-10.0.0 /path/to/geant4.10.00/examples/basic/B1
```

```
-- The C compiler identification is GNU
-- The CXX compiler identification is GNU
-- Check for working C compiler: /usr/bin/gcc
-- Check for working C compiler: /usr/bin/gcc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - 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
-- Configuring done
-- Generating done
-- Build files have been written to: /home/you/B1-build
```

```
$ ls
CMakeCache.txt      exampleB1.in      init_vis.mac      run2.mac
CMakeFiles          exampleB1.out     Makefile          vis.mac
cmake_install.cmake  init.mac          run1.mac
```

```
$ make -jN
```

```
$ ./exampleB1
*****
Geant4 version Name: geant4-10-00-ref-00 [MT] (6-December-2013)
<< in Multi-threaded mode >>
Copyright : Geant4 Collaboration
Reference : NIM A 506 (2003), 250-303
WWW : http://cern.ch/geant4
*****

<<< Reference Physics List QBBC
Checking overlaps for volume Envelope ... OK!
Checking overlaps for volume Shapel ... OK!
Checking overlaps for volume Shape2 ... OK!
User-derived detector construction class does not implement
ConstructSDandField method: i.e. workers will not have SD and fields!
The user can safely ignore this message if (s)he has no sensitive
detector or field in her/his application.
### Adding tracking cuts for neutron TimeCut(ns)= 10000 KinEnergyCut(MeV)= 0
Visualization Manager instantiating with verbosity "warnings (3)"...
Visualization Manager initialising...
Registering graphics systems...
```


Geant4 Örnekleri

../geant4.9.6-install/share/Geant4-9.6.0/examples altında:

- Temel (basic) örnekler
 - ▶ Geant4 uygulamalarının en tipik örnekleri
- Yeni başlayan (novice) örnekleri
 - ▶ Etkileşmeyen parçacıklardan oldukça karmaşık algıç benzetimlerine kadar değişen uygulamalar
- Genişletilmiş (extended) örnekler (Geant4'ün özel uygulamalarının gösterilmesi)
 - ▶ Elektromanyetik
 - ▶ Analiz
 - ▶ Önyargılama
 - ▶ Görüntüleme
 - ▶
- İleri (advanced) örnekler (gerçek deney düzeneklerinin ya da cihazların benzetimi)
 - ▶ Brachytherapy
 - ▶ Gammaray_telescope
 - ▶ Medical_linac
 - ▶ Hadrontherapy

10. Examples

10.1. Introduction

10.2. Basic Examples

10.2.1. Basic Examples Summary

10.2.2. Basic Examples Macros

10.2.3. Multi-threading

10.2.4. Example B1

10.2.5. Example B2

10.2.6. Example B3

10.2.7. Example B4

10.2.8. Example B5

10.3. Extended Examples

10.3.1. Extended Example Summary

10.4. Advanced Examples

10.5. Novice Examples

<http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/ForApplicationDeveloper/html/>

Malzeme Tanımı

Malzeme Tanımı

- Üç tür malzeme tanımlanabilir:
 - ▶ isotop → G4Isotope
 - ▶ element → G4Element
 - ▶ molekül → G4Material
 - ▶ alaşım ve bileşim → G4Material
- Bu malzemelerin özellikleri olabilir
 - ▶ Sıcaklık, basınç, durum (katı, sıvı, gaz), **yoğunluk**
- **G4Isotope** ve **G4Element** atomların özelliğini betimler
 - ▶ Atom sayısı, nükleon sayısı, molar kütle, kabuk enerjileri, atom başına tesir kesiti, ...
- **G4Material** maddenin mikroskobik özelliklerini betimler:
 - ▶ sıcaklık, basınç, durum, yoğunluk
 - ▶ Radyasyon uzunluğu, soğurma uzunluğu, ...

G4Material izleme, geometri ve fizik sınıfları tarafından kullanılır.

Elementler ve İzotoplar

Izotoplar elementlerden oluşturulabilir:

```
G4Isotope (const G4String& name,  
           G4int      z,      // atomic number  
           G4int      n,      // number of nucleons  
           G4double   a );   // mass of mole
```

Elementler aşağıdaki gibi oluşturulup izotopa eklenir:

```
G4Element (const G4String& name,  
           const G4String& symbol, // element symbol  
           G4int      nIso ); // n. of isotopes  
G4Element::AddIsotope (G4Isotope* iso, // isotope  
                       G4double relAbund);  
                        // fraction of atoms  
                        // per volume
```

Bir Elementten ve Moleküllerden Oluşmuş Malzeme

Tek elementli malzeme:

```
G4double z, a, density
density = 1.390*g/cm3;
a = 39.95*g/mole;
G4Material* lAr =
  new G4Material("liquidArgon", z=18, a, density);
```

Birden fazla elementten oluşmuş molekül
(atom sayısı ile birleştirme):

```
a = 1.01*g/mole;
G4Element* elH =
  new G4Element("Hydrogen", symbol="H", z=1., a);
a = 16.00*g/mole;
G4Element* elO =
  new G4Element("Oxygen", symbol="O", z=8., a);
density = 1.000*g/cm3;
G4Material* H2O =
  new G4Material("Water", density, ncomponents=2);
H2O->AddElement(elH, natoms=2);
H2O->AddElement(elO, natoms=1);
```

Alařım ve Karıřımlar

Karıřım: kesirli birleřtirme

```
a = 14.01*g/mole;  
G4Element* elN = new G4Element(name="Nitrogen",symbol="N",z= 7.,a) ;  
a = 16.00*g/mole;  
G4Element* elO = new G4Element(name="Oxygen",symbol="O",z= 8.,a) ;  
density = 1.290*mg/cm3;  
G4Material* Air = new G4Material(name="Air",density,ncomponents=2) ;  
Air->AddElement(elN, 70.0*perCent) ;  
Air->AddElement(elO, 30.0*perCent) ;
```

Alařım malzemelerinin birleřimi

```
G4Element* elC = ...; // define "carbon" element  
G4Material* SiO2 = ...; // define "quartz" material  
G4Material* H2O = ...; // define "water" material  
density = 0.200*g/cm3;  
G4Material* Aerog = new G4Material("Aerogel",density,ncomponents=3) ;  
Aerog->AddMaterial(SiO2,fractionmass=62.5*perCent) ;  
Aerog->AddMaterial(H2O ,fractionmass=37.4*perCent) ;  
Aerog->AddElement (elC ,fractionmass= 0.1*perCent) ;
```

NIST Malzeme Veritabanı

<http://physics.nist.gov/PhysRefData>

Elementleri ve malzemeleri ön tanımlamadan doğrudan NIST Manager ile veritabanından kullanabilirsiniz:

```
G4NistManager* manager = G4NistManager::Instance();  
G4Material* H2O = manager->FindOrBuildMaterial("G4_WATER");
```

3000'den fazla malzeme veritabanında bulunmaktadır:

8. Geant4 Material Database

8.1. Pure Materials

8.2. NIST Compounds

8.3. HEP and Nuclear Materials

8.4. Space (ISS) Materials

8.5. Bio-Chemical Materials

Geometri Tanımı

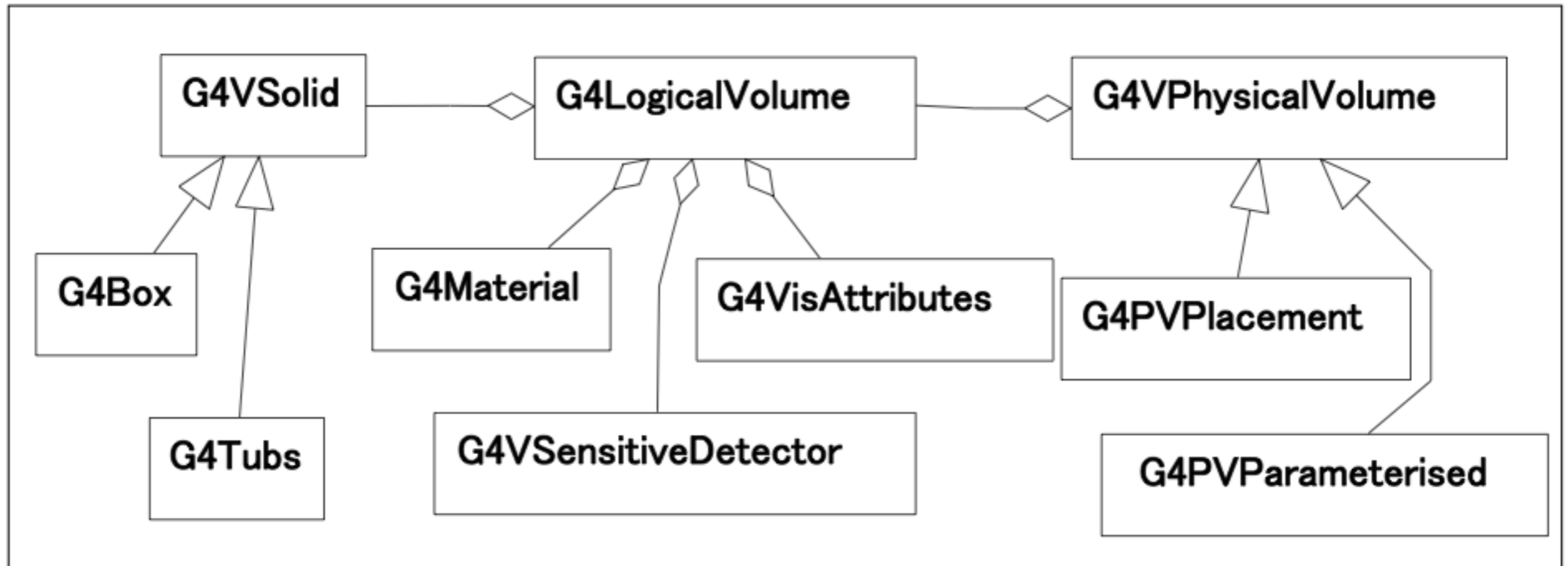
Algıç Tanımı

- Algıç geometrisi bir kaç hacimden oluşur.
- Bu hacimlerin en büyüğü World hacmi olarak isimlendirilir
 - ▶ Diğer tüm hacimleri içine almalıdır.
- Algıcın somut sınıfı G4VUserDetectorConstruction soyut baz sınıfından türetilir.
- Construct() metodunun uygulanması:
- Modüler biçimde algıcın her alt parçası için yazılır:
 - ▶ Geometriyi tanımlamak için gerekli şekil ve katılar tanımlanır
 - ▶ Gerekli tüm malzemeler oluşturulur
 - ▶ Algıç geometisindeki hacimler oluşturulur ve yerlerine yerleştirilir
 - ▶ Hassas algıçlar tanımlanır ve hangi algıç hacimlerinin bunlarla eşleştirileceğine karar verilir
 - ▶ Manyetik alan olacak ise algıcın hangi bölgesine ekleneceğine karar verilir
 - ▶ Algıç elemanları için görüntüleme özellikleri tanımlanır (renk, görünürlük vb.)

Algıç Geometrisinin Bileşenleri

Kavramsal olarak üç katmandan oluşur:

- **G4VSolid**: şekil, boyut
- **G4VLogicalVolume**: malzeme, manyetik alan, hassaslık, görüntüleme özellikleri, yavru hacimler
- **G4VPhysicalVolume**: konum, dönme, kopyalama



Algıç Geometrisi Tanımlama

En basit örnek: examples/basic/B1

```
G4Box* solidWorld =
  new G4Box("World",           //its name
            0.5*world_sizeXY, 0.5*world_sizeXY, 0.5*world_sizeZ); //its size

G4LogicalVolume* logicWorld =
  new G4LogicalVolume(solidWorld, //its solid
                      world_mat,  //its material
                      "World");  //its name

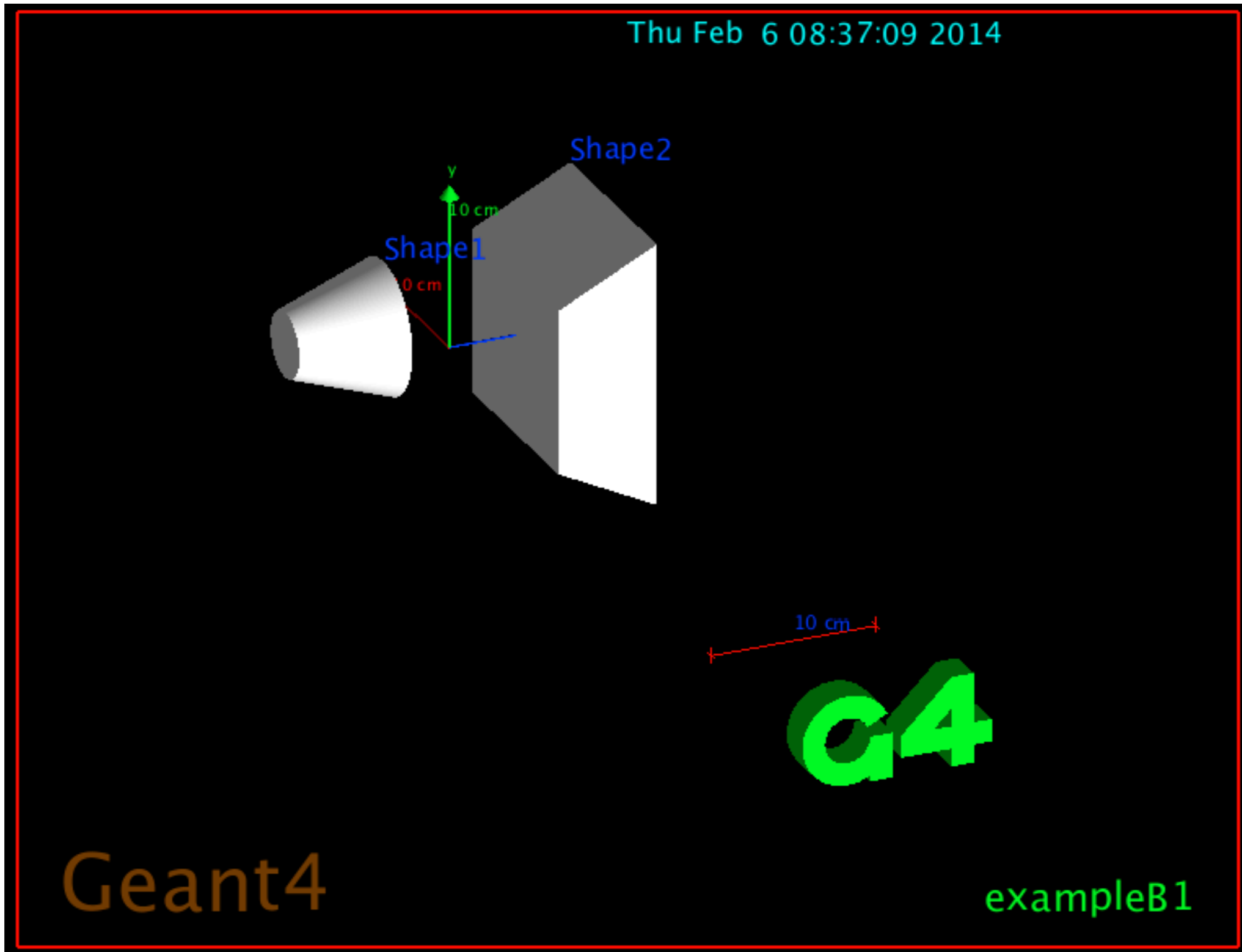
G4VPhysicalVolume* physWorld =
  new G4PVPlacement(0,           //no rotation
                   G4ThreeVector(), //at (0,0,0)
                   logicWorld,  //its logical volume
                   "World",     //its name
                   0,           //its mother volume
                   false,       //no boolean operation
                   0,           //copy number
                   checkOverlaps); //overlaps checking

G4Box* solidEnv =
  new G4Box("Envelope",       //its name
            0.5*env_sizeXY, 0.5*env_sizeXY, 0.5*env_sizeZ); //its size

G4LogicalVolume* logicEnv =
  new G4LogicalVolume(solidEnv, //its solid
                      env_mat,  //its material
                      "Envelope"); //its name

new G4PVPlacement(0,           //no rotation
                 G4ThreeVector(), //at (0,0,0)
                 logicEnv,     //its logical volume
                 "Envelope",   //its name
                 logicWorld,   //its mother volume
                 false,        //no boolean operation
                 0,            //copy number
                 checkOverlaps); //overlaps checking
```

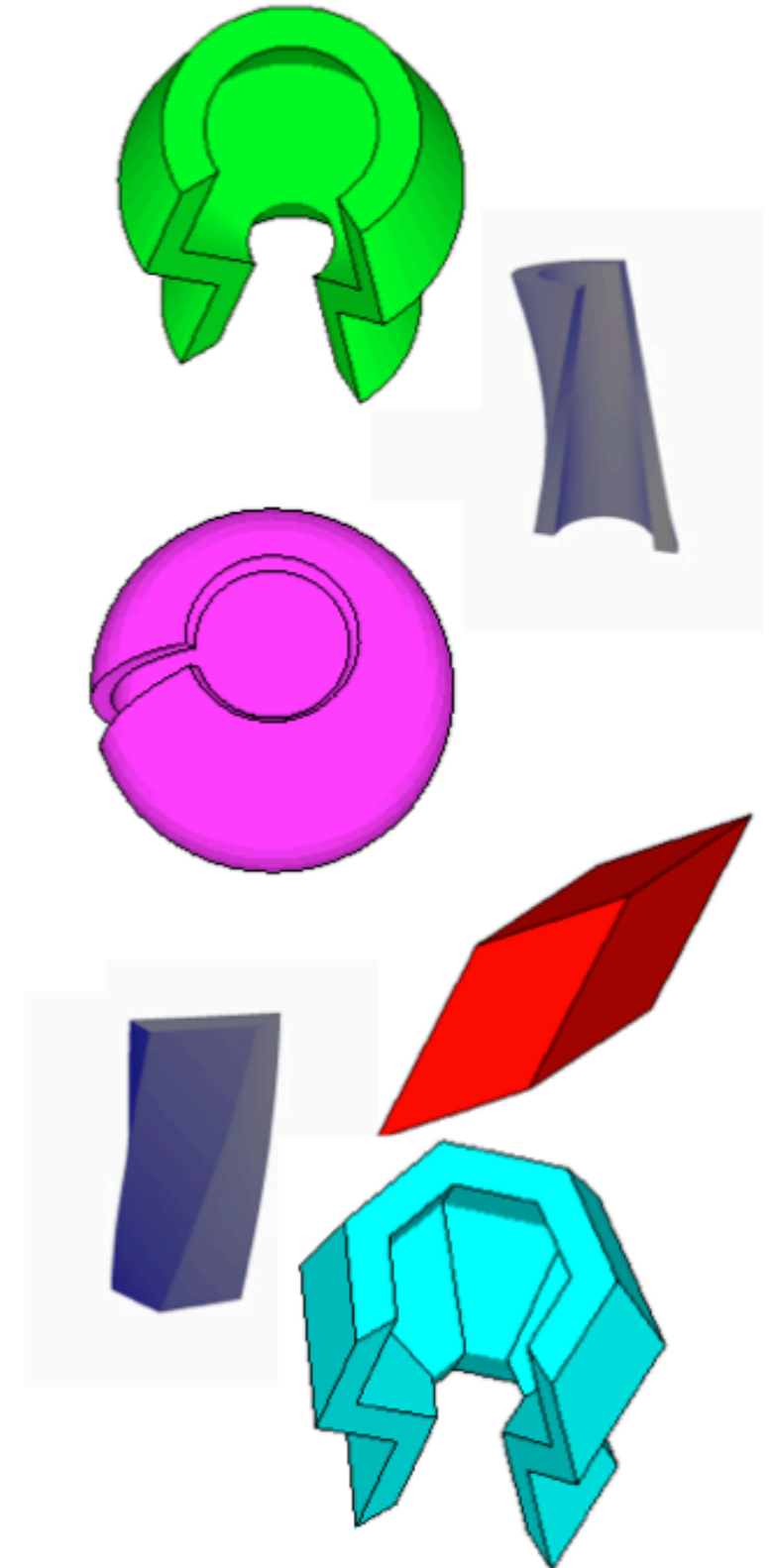
/example/basic/B1



Katılar

Geant4'te katılar farklı yollarla tanımlanabilir:

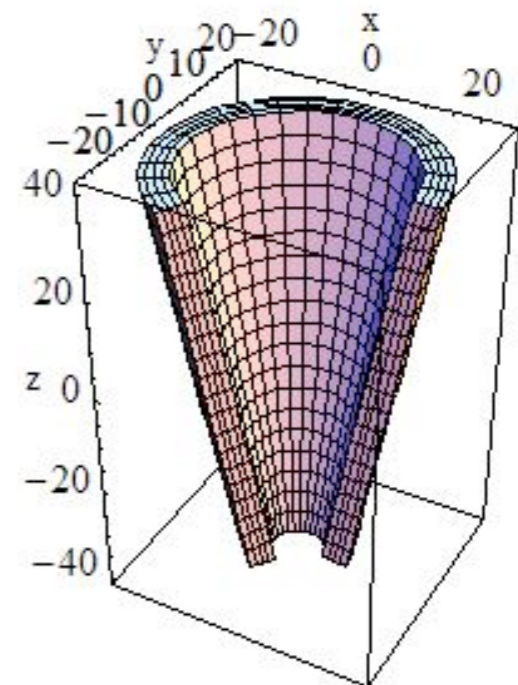
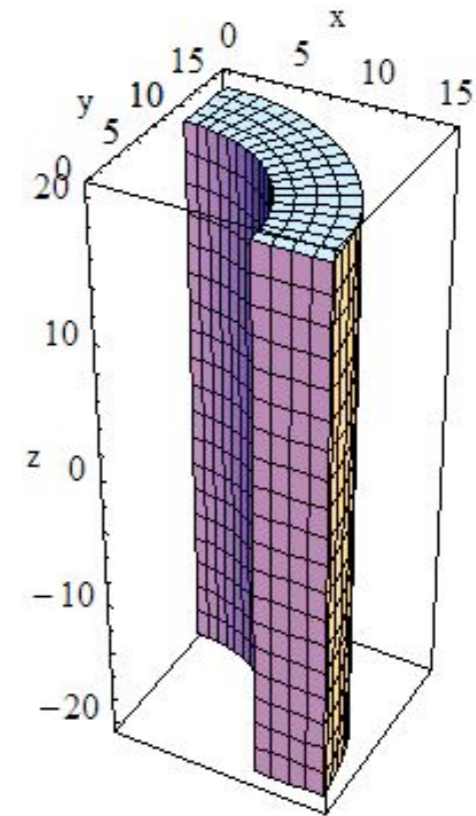
- CSG (Constructed Solid Geometry) katılar
 - ▶ G4Box, G4Tubs, G4Cons, G4Trd, ...
- Özel katılar (CSG tarzında)
 - ▶ G4Polycone, G4Polyhedra, G4Hype, ...
 - ▶ G4TwistedTubs, G4TwistedTrap, ...
- BREP (BoundaryREPresented) katılar
 - ▶ G4BREP-solidPolycone, G4BSplineSurface, ...
- Boolean katılar
 - ▶ G4UnionSolid, G4SubtractionSolid, ...



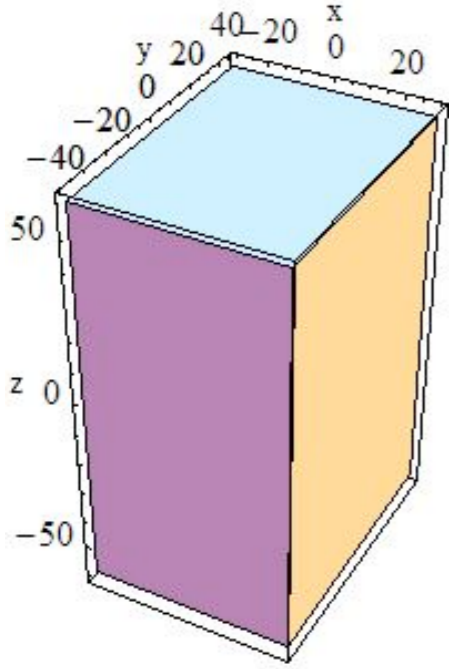
CSG: G4Tubs, G4Cons

```
G4Tubs (const G4String& pname, // name
         G4double pRmin, // inner radius
         G4double pRmax, // outer radius
         G4double pDz, // Z half length
         G4double pSphi, // starting Phi
         G4double pDphi); // segment angle
```

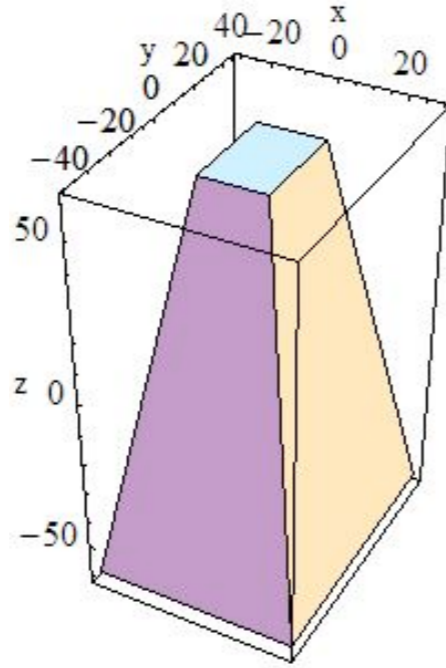
```
G4Cons (const G4String& pname, // name
         G4double pRmin1, // inner radius -pDz
         G4double pRmax1, // outer radius -pDz
         G4double pRmin2, // inner radius +pDz
         G4double pRmax2, // outer radius +pDz
         G4double pDz, // Z half length
         G4double pSphi, // starting Phi
         G4double pDphi); // segment angle
```



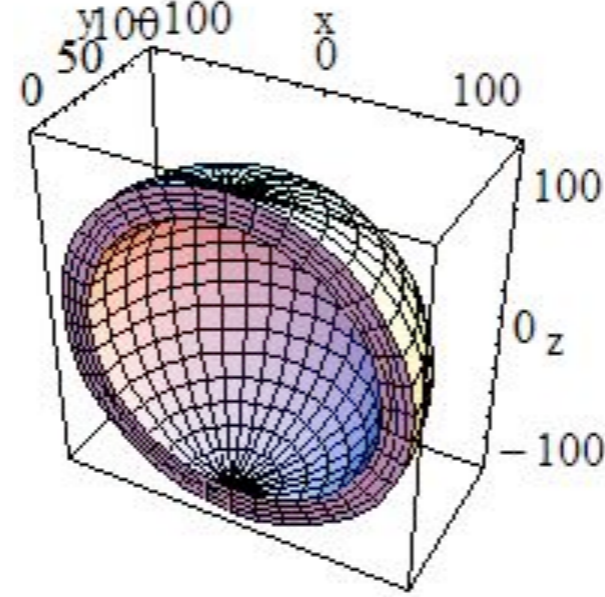
Diğer CSG Katılar



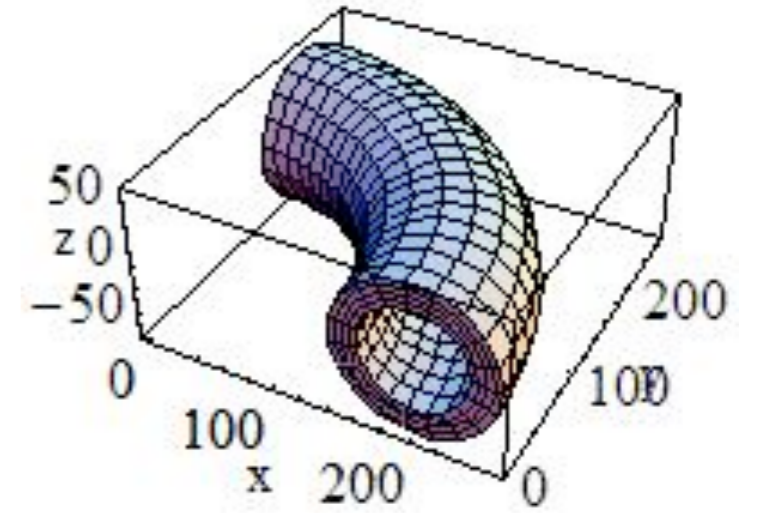
G4Box



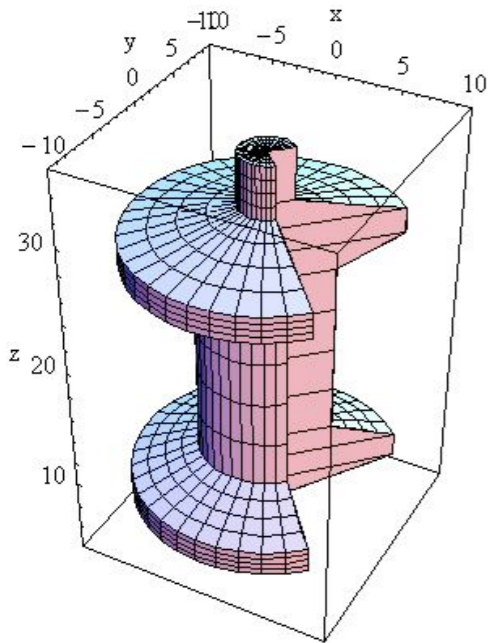
G4Trd



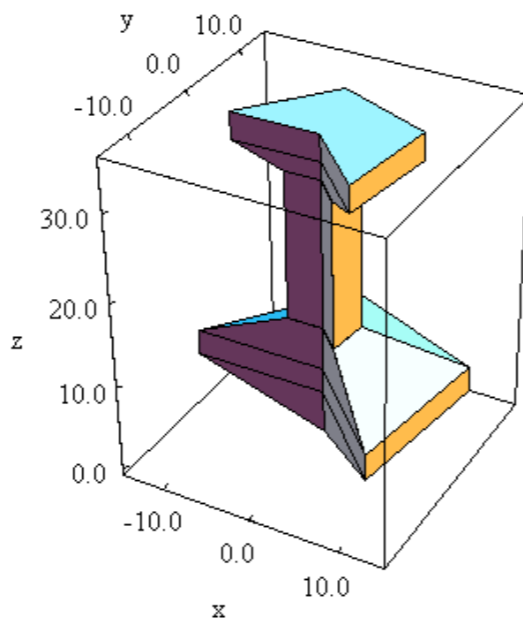
G4Sphere



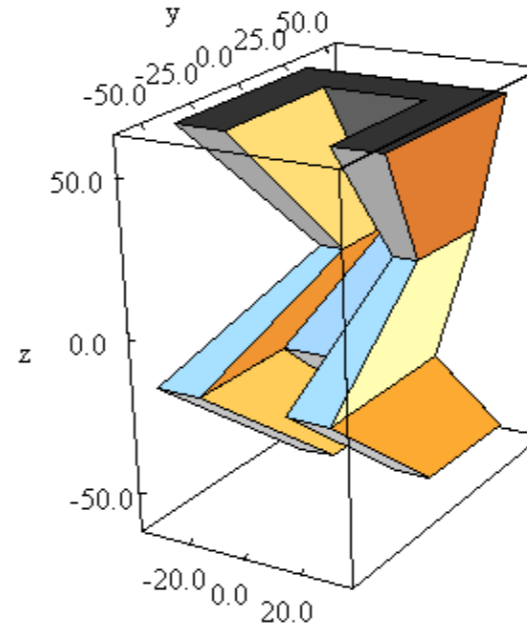
G4Torus



G4Polycone



G4Polyhedra



G4ExtrudeSolid

Tüm ayrıntılar için:
[Application Developers Guide Bölüm 4.1.2](#)

Birincil Paracık Demeti Üretilmesi

Kullanıcının Birincil Parçacık Üretmesi

- Geant4'te G4VPrimaryGenerator sınıfının üç somut sınıfı vardır:
 - ▶ G4ParticleGun
 - ▶ G4GeneralParticleSource
 - ▶ G4HEPEvtInterface, G4HEPMCInterface
- **G4ParticleGun**
 - ▶ Tipik deneylerde kullanılan demet parçacıklarını benzetmek için kullanılabilir.
 - ▶ Birincil köşe yüzeydeki ya da 3D nesnelerin hacimleri içinde bir yerde noktasal kaynaklardan gelişigüzel olarak seçilebilir. Birincil parçacığın momentum yönü ve kinetik enerjisi gelişigüzel halde olabilir.
- **G4GeneralParticleSource**
 - ▶ Genellikle uzay uygulamaları için kullanılır.
 - ▶ Uygulanması basittir ve UI komutları ile koşu başında değerleri ayarlanabilir.
- **G4HEPEvtInterface**
 - ▶ Deneylere özel birincil parçacık üretimi için kullanılabilir.
 - ▶ Pek çok YEF (FORTRAN) fizik üreticinin kullandığı /HEPEVT/ yaygın blok için uygundur
 - ▶ ASCII dosyası girdisi kullanabilir.
- **G4HEPMCInterface**
 - ▶ Deneylere özel birincil parçacık üretimi için kullanılabilir.
 - ▶ Bazı (C++) YEF fizik üreticilerinin kullandığı HepMC formatı için uygundur
 - ▶ ASCII girdi dosyası ya da HepMC ile doğrudan bağlantılama yolu ile kullanılabilir.

Örnek:G4ParticleGun

```
#include "G4ParticleGun.hh"
G4ParticleGun particleGun;
void myPrimaryGeneratorAction::GeneratePrimaries (G4Event* anEvent)
{
    particleGun->SetParticleDefinition (G4Electron::Definition());
    particleGun->SetParticleMomentum (G4ThreeVector (1.0,0,0));
    particleGun->SetParticleEnergy (100.0*keV);
    particleGun->GeneratePrimaryVertex (anEvent);
}
```

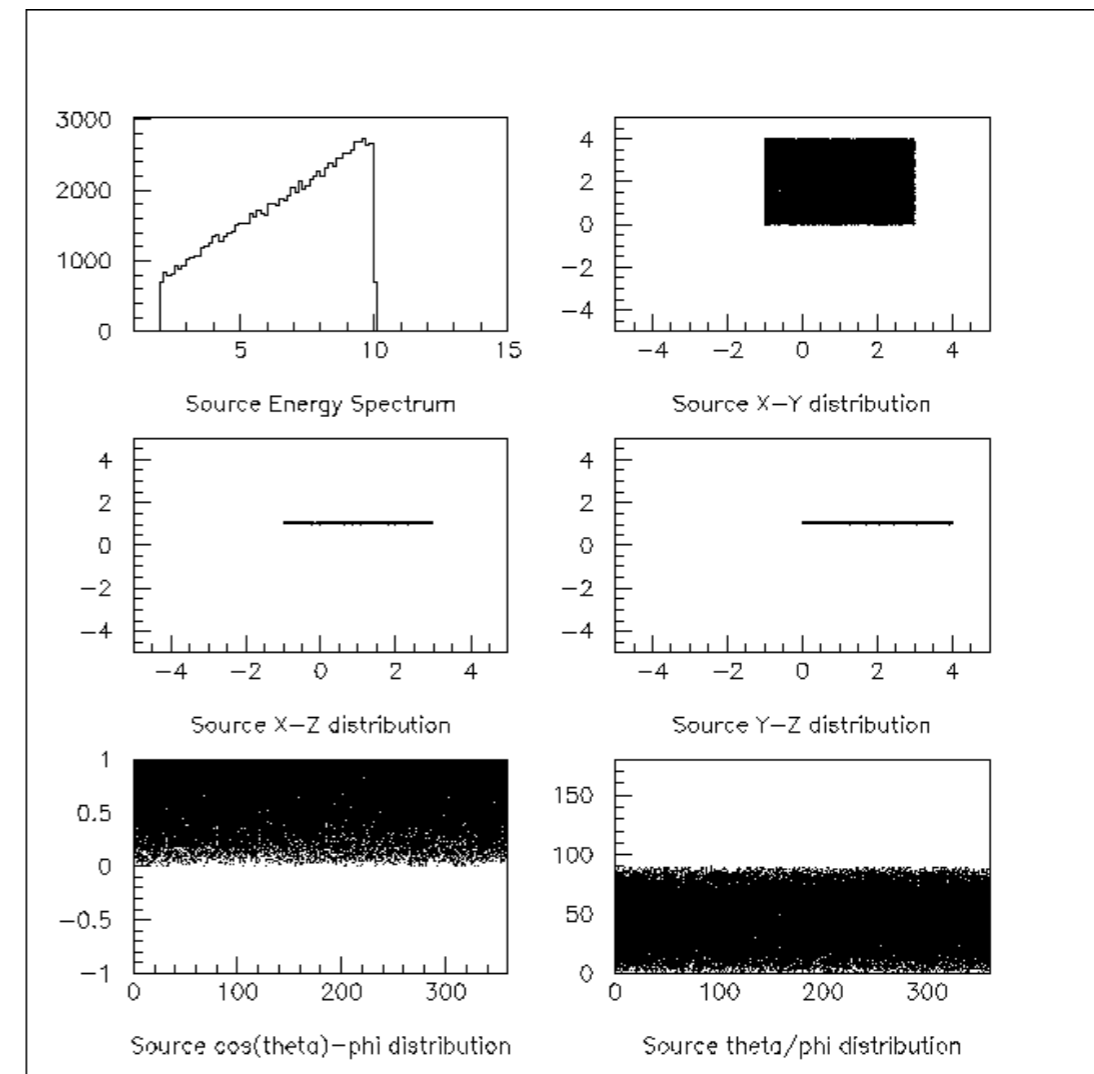
Kullanılabilir Set Metodları:

```
void SetParticleDefinition (G4ParticleDefinition*)
void SetParticleMomentum (G4ParticleMomentum)
void SetParticleMomentumDirection (G4ThreeVector)
void SetParticleEnergy (G4double)
void SetParticleTime (G4double)
void SetParticlePosition (G4ThreeVector)
void SetParticlePolarization (G4ThreeVector)
void SetNumberOfParticles (G4int)
```

Örnek:G4GeneralParticleSource

```
myPrimaryGeneratorAction::myPrimaryGeneratorAction()  
{ generator = new G4GeneralParticleSource; }  
  
void myPrimaryGeneratorAction::GeneratePrimaries(G4Event* anEvent)  
{ generator->GeneratePrimaryVertex(anEvent); }
```

```
/gps/particle gamma  
/gps/pos/type Plane  
/gps/pos/shape Square  
/gps/pos/centre 1. 2. 1. cm  
/gps/pos/halfx 2. cm  
/gps/pos/halfy 2. cm  
/gps/ang/type cos  
/gps/ene/type Lin  
/gps/ene/min 2. MeV  
/gps/ene/max 10. MeV  
/gps/ene/gradient 1.  
/gps/ene/intercept 1.
```



Öneriler

G4ParticleGun

- Basit ve sadedir
- Tane tane parçacık atar.
- Kullanması/kontrol etmesi kolaydır
 - ▶ Herbir iz için ya da olay için değeri değiştirmek için Set () metodlarını kullanabilirsiniz.

G4GeneralParticleSource

- Güçlüdür
- UI komutları ile kontrol edilir
 - ▶ Set () metodları ile kontrol etmek neredeyse mümkün değildir.
 - ▶ Bir hacmin yüzeyinden parçacık gönderme imkanı verir.
 - ▶ Kullanıcının verdiği bir dağılımla (histogram) parçacıkların kinetik enerjisini, konumunu ve/veya yönünü gelişigüzel olarak değiştirmek mümkündür.

- Bir hacmin yüzeyinden, içeri doğru ya da dışarı doğru, birincil parçacık gönderekseniz G4GeneralParticleSource kullanın.
- Düz ya da basit bir Gaussian dağılım olmayan karmaşık bir birincil parçacık dağılımına ihtiyacınız varsa G4GeneralParticleSource kullanın.
- Diğer durumlarda G4ParticleGun kullanabilirsiniz.

Parçacıklar ve Süreçler

Geant4'de Fizik Süreçleri

- Geant4 fizik süreçlerinin benzetiminde tek bir bütünleyici yaklaşım yerine tane tane (parça parça) olan bir yaklaşım kullanır
 - ▶ birbiri ile (çoğu zaman) ilişkili olmayan fizik bileşenleri (süreçler) sunar
 - ▶ kullanıcı algıci parça parça inşa ettiği gibi bu fizik bileşenlerini de parça parça birleştirerek uyarladığı fizik listesine ekler
- İstisnalar
 - ▶ bazı elektromanyetik süreçler birlikte kullanılmalıdır
 - örneğin: ionizasyon ve frenleme ışınımı: her ikisi de, yüklü parçacığın erimini belirleyen enerji kaybına katkıda bulunurlar.

Geant4'ün Sağladığı Fizik Süreçleri

- EM Etkileşmeler
 - ▶ $\sim 10^3$ eV'den $\sim 10^{15}$ eV'e kadar geçerli olan “standart” süreçler
 - ▶ ~ 250 eV'den $\sim 10^{15}$ eV'e kadar geçerli olan “düşük enerji” süreçler
 - ▶ optik fotonlar
- Zayıf Etkileşmeler
 - ▶ atomaltı parçacıkların bozunumu
 - ▶ çekirdeğin radyoaktif bozunumu
- Güçlü Etkileşmeler
 - ▶ ~ 0 eV'den $\sim 10^{12}$ eV'e kadar geçerli tüm güçlü etkileşmeler
 - ▶ $\sim 10^6$ eV'den $\sim 10^{12}$ eV'e kadar geçerli elektro- ve gamma-nükleer etkileşmeler
- Parametrize edilmiş ya da hızlı benzetim fiziği

G4VUserPhysicsList

- Tüm fizik süreçleri bu sınıftan türetilmelidir
 - ▶ ardından koşu yöneticisine (run manager) kaydedilmelidir
- Kullanıcı ConstructParticle(), ConstructProcess() and SetCuts() metodlarını kendisi yazmalıdır.

Örnek:

```
37 #ifndef PhysicsList_h
38 #define PhysicsList_h 1
39
40 #include "G4VUserPhysicsList.hh"
41 #include "globals.hh"
42
43 //....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....
44
45 class PhysicsList: public G4VUserPhysicsList
46 {
47 public:
48     PhysicsList();
49     virtual ~PhysicsList();
50
51     // Construct particle and physics
52     virtual void ConstructParticle();
53     virtual void ConstructProcess();
54
55     virtual void SetCuts();
56
57 private:
58
59     // these methods Construct physics processes and register them
60     void ConstructDecay();
61     void ConstructEM();
62 };
63
64 //....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....
65
66 #endif
```


G4VUserPhysicsList:Gerekli Metdolar

- **ConstructParticle()** – benzetimde ihtiyaç duyacağınız tüm parçacıkları seçip burada tanımlayın.
- **ConstructProcess()** – her parçacık için, benzetiminizde önemli olduğunu düşündüğünüz tüm fizik süreçlerini atayın
- Fizik süreci nedir?
 - ▶ bir parçacığın madde ile nasıl etkileşeceğini tanımlayan bir sınıftır
- **SetCuts()** – ikincil parçacık üretimi için erim kesme değerlerini tanımlayın
- Erim kesme nedir?
 - ▶ parçacık üretiminde ne kadar düşük enerjilere gidileceğini belirleyen sınırdır

ConstructParticle()

```
103 void PhysicsList::ConstructParticle()
104 {
105     // pseudo-particles
106     G4Geantino::GeantinoDefinition();
107     G4ChargedGeantino::ChargedGeantinoDefinition();
108
109     // gamma
110     G4Gamma::GammaDefinition();
111
112     // optical photon
113     G4OpticalPhoton::OpticalPhotonDefinition();
114
115     // leptons
116     G4Electron::ElectronDefinition();
117     G4Positron::PositronDefinition();
118     G4MuonPlus::MuonPlusDefinition();
119     G4MuonMinus::MuonMinusDefinition();
120
121     G4NeutrinoE::NeutrinoEDefinition();
122     G4AntiNeutrinoE::AntiNeutrinoEDefinition();
123     G4NeutrinoMu::NeutrinoMuDefinition();
124     G4AntiNeutrinoMu::AntiNeutrinoMuDefinition();
125
126     // mesons
127     G4MesonConstructor mConstructor;
128     mConstructor.ConstructParticle();
129
130     // barions
131     G4BaryonConstructor bConstructor;
132     bConstructor.ConstructParticle();
133
134     // ions
135     G4IonConstructor iConstructor;
136     iConstructor.ConstructParticle();
137
138     // Required by MT even if ion physics not used
139     G4GenericIon::GenericIonDefinition();
140 }
```

ConstructProcess()

```
126 void PhysicsList::ConstructProcess()  
127 {  
128     AddTransportation();  
129     ConstructEM();  
130     ConstructGeneral();  
131 }
```

ConstructEM()

```
135 void PhysicsList::ConstructEM()
136 {
137     theParticleIterator->reset();
138     while( (*theParticleIterator)() ){
139         G4ParticleDefinition* particle = theParticleIterator->value();
140         G4ProcessManager* pmanager = particle->GetProcessManager();
141         G4String particleName = particle->GetParticleName();
142
143         if (particleName == "gamma") {
144             // gamma
145             pmanager->AddDiscreteProcess(new G4PhotoElectricEffect);
146             pmanager->AddDiscreteProcess(new G4ComptonScattering);
147             pmanager->AddDiscreteProcess(new G4GammaConversion);
148             pmanager->AddDiscreteProcess(new G4RayleighScattering);
149
150         } else if (particleName == "e-") {
151             //electron
152             pmanager->AddProcess(new G4eMultipleScattering, -1, 1, 1);
153             pmanager->AddProcess(new G4eIonisation, -1, 2, 2);
154             pmanager->AddProcess(new G4eBremsstrahlung, -1, 3, 3);
155             if (fSRType) {
156                 pmanager->AddProcess(new G4SynchrotronRadiation, -1,-1, 4);
157             } else {
158                 pmanager->AddProcess(new G4SynchrotronRadiationInMat, -1,-1, 4);
159             }
160             pmanager->AddProcess(new G4StepLimiter, -1,-1, 5);
161
162         } else if (particleName == "e+") {
```

ConstructGeneral()

```
191 void PhysicsList::ConstructGeneral()  
192 {  
193     // Add Decay Process  
194     G4Decay* theDecayProcess = new G4Decay();  
195     theParticleIterator->reset();  
196     while ((*theParticleIterator)()){  
197         G4ParticleDefinition* particle = theParticleIterator->value();  
198         G4ProcessManager* pmanager = particle->GetProcessManager();  
199         if (theDecayProcess->IsApplicable(*particle)) {  
200             pmanager ->AddProcess(theDecayProcess);  
201             // set ordering for PostStepDoIt and AtRestDoIt  
202             pmanager ->SetProcessOrdering(theDecayProcess, idxPostStep);  
203             pmanager ->SetProcessOrdering(theDecayProcess, idxAtRest);  
204         }  
205     }  
206 }
```

SetCuts()

```
210 void PhysicsList::SetCuts()  
211 {  
212     if (verboseLevel >0){  
213         G4cout << "CutLength : " << G4BestUnit(defaultCutValue, "Length") << G4endl;  
214     }  
215  
216     // set cut values for gamma at first and for e- second and next for e+,  
217     // because some processes for e+/e- need cut values for gamma  
218     SetCutValue(defaultCutValue, "gamma");  
219     SetCutValue(defaultCutValue, "e-");  
220     SetCutValue(defaultCutValue, "e+");  
221  
222     if (verboseLevel>0) DumpCutValuesTable();  
223 }
```

Fizik Süreçleri: Kısa Özet

- Bir uygulama için gerekli olan tüm parçacıklar, fizik süreçleri ve üretim kesimleri fizik listesine eklenmelidir.
- Kullanıcıların kendi fizik sınıflarını türetebilecekleri iki fizik listesi sınıfı vardır:
 - ▶ G4VUserPhysicsList – görece basit fizik listeleri için
 - ▶ G4VModularPhysicsList – daha ayrıntılı fizik listeleri için
- Geant4'ün içinde önceden hazırlanmış fizik listeleri vardır:
 - ▶ elektromanyetik fizik listesi
 - ▶ elektromanyetik + hadronik fizik listeleri

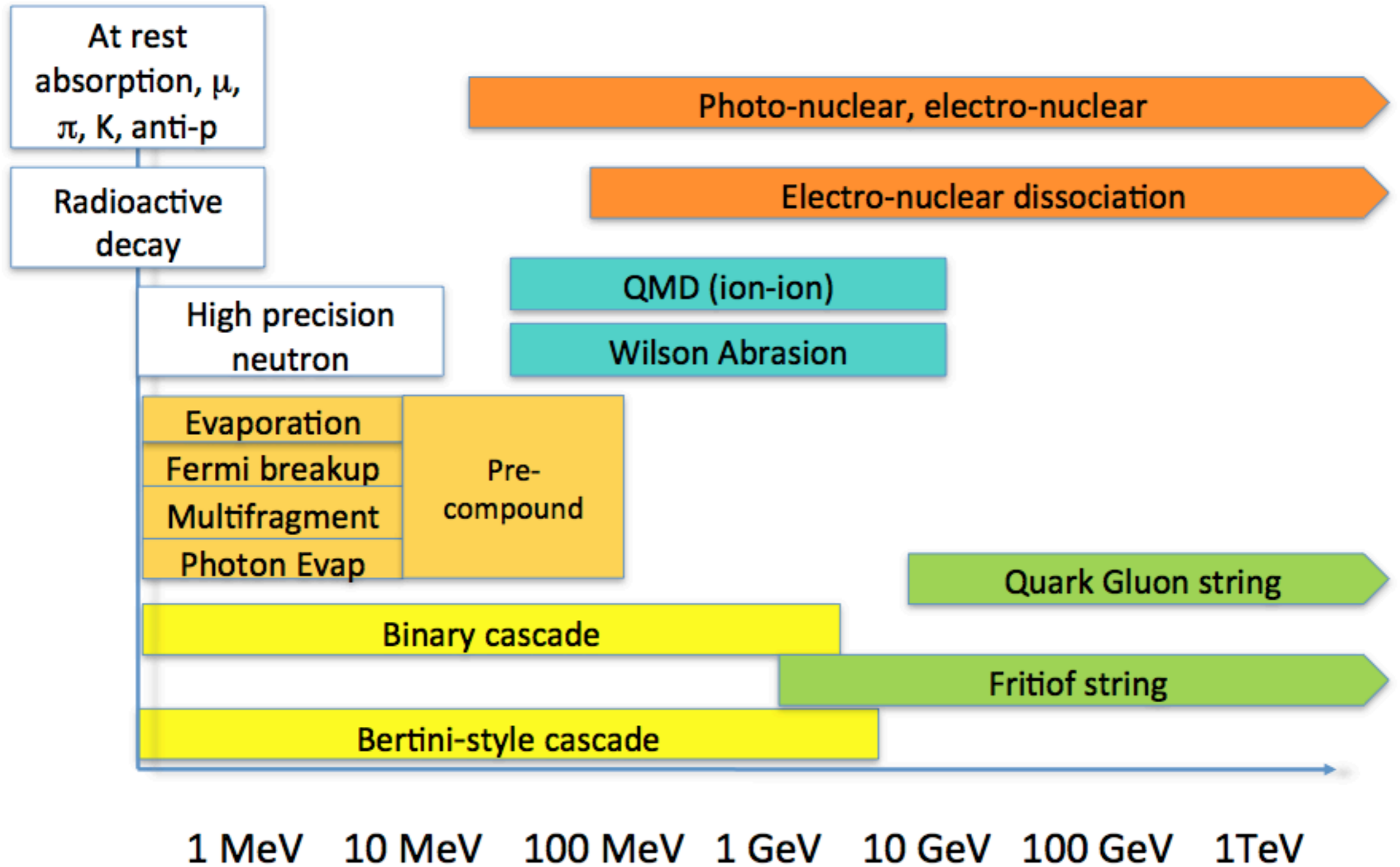
Kullanıcı doğru fiziği seçme konusunda hassas olmalıdır.

Geant4 EM packages

- **Standard**
 - γ , e up to 100 TeV
 - hadrons up to 100 TeV
 - ions up to 100 TeV
- **Muons**
 - up to 1 PeV
 - energy loss propagator
- **X-rays**
 - X-ray and optical photon production proc.
- **High-energy**
 - processes at high energy ($E > 10\text{GeV}$)
 - physics for exotic particles
- **Polarisation**
 - simulation of polarized beams
- **Optical**
 - optical photon interactions
- **Low-energy**
 - Livermore library γ , e- from 250 eV up to 1 GeV
 - Livermore library based polarized processes
 - PENELOPE code rewrite , γ , e- , e+ from 100 eV up to 1 GeV (2008 version)
 - hadrons and ions up to 1 GeV
 - atomic de-excitation (fluorescence + Auger)
- **Geant4-DNA**
 - microdosimetry models for radiobiology (Geant4-DNA project) from 0.025 eV to 10 MeV
- **Adjoint**
 - New sub-library for reverse Monte Carlo simulation from the detector of interest back to source of radiation
- **Utils**
 - general EM interfaces

Slayt: Sebastien Inceriti

Partial Hadronic Model Inventory



Slayt: Dennis Wright

Diğer Kaynaklar

Kullanılan Diğer Kaynaklar:

[https://agenda.infn.it/getFile.py/access?
sessionId=17&resId=0&materialId=0&confId=5699](https://agenda.infn.it/getFile.py/access?sessionId=17&resId=0&materialId=0&confId=5699)

[https://agenda.infn.it/getFile.py/access?
sessionId=5&resId=0&materialId=0&confId=5699](https://agenda.infn.it/getFile.py/access?sessionId=5&resId=0&materialId=0&confId=5699)