

Geant4 ile Detektör Tasarımı ve Benzetimi - I

Doç. Dr. Sinan KUDAY

Ankara Üniversitesi Fizik Bölümü
skuday@ankara.edu.tr

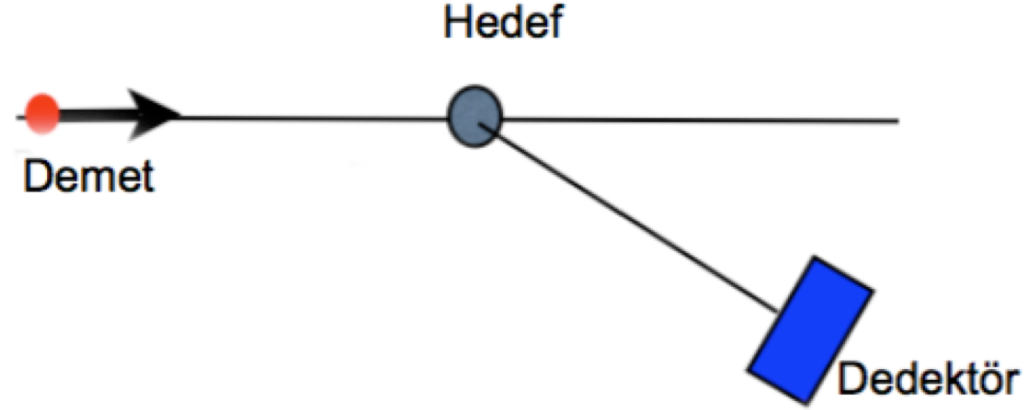
- 1. Geant4 Geçmişi ve Kullanım Alanları**
- 2. Geant4 Kurulumu & Terminolojisi**
- 3. Kullanıcı Arayüzü & Temel Komutlar**
- 4. Basit Uygulama**

Kısım 1:

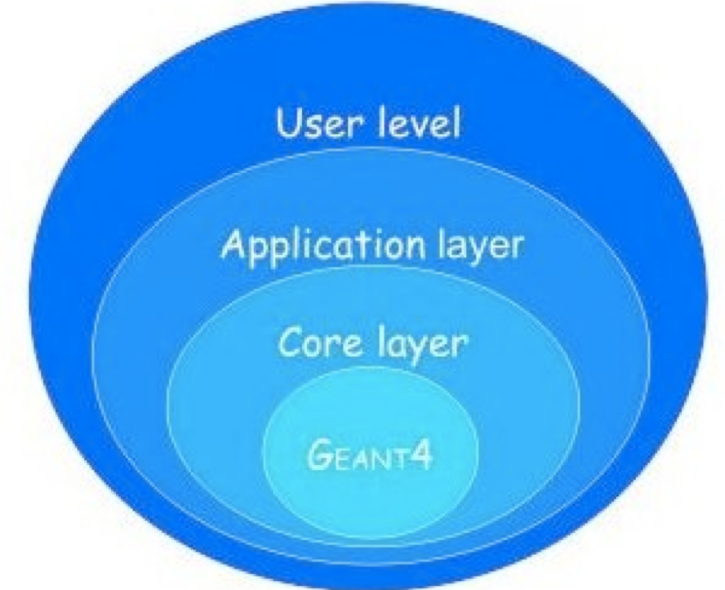
Geant4 Geçmişi ve Kullanım Alanları

- Geant4, C++ polimorfizmi ve nesne yönelimli programlaması sayesinde aşağıdaki başlıklarda kullanıcılara kendi tanımlamalarını yapmaları sağlar.
- Geant4 ile doğruluğu ispatlanmış disiplinler-arası örnekler de yapılmıştır. (*Bknz: ilerleyen sayfalar*)

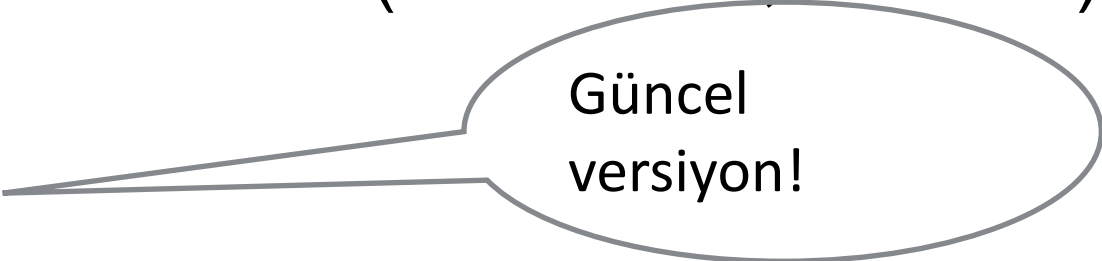
- Demet
- Hedef
- Dedektör
- Etkileşim Alanları
- Fizik Etkileşimleri ve Algoritmaları
- Analiz Yöntemleri



**Root Analysis
Framework**



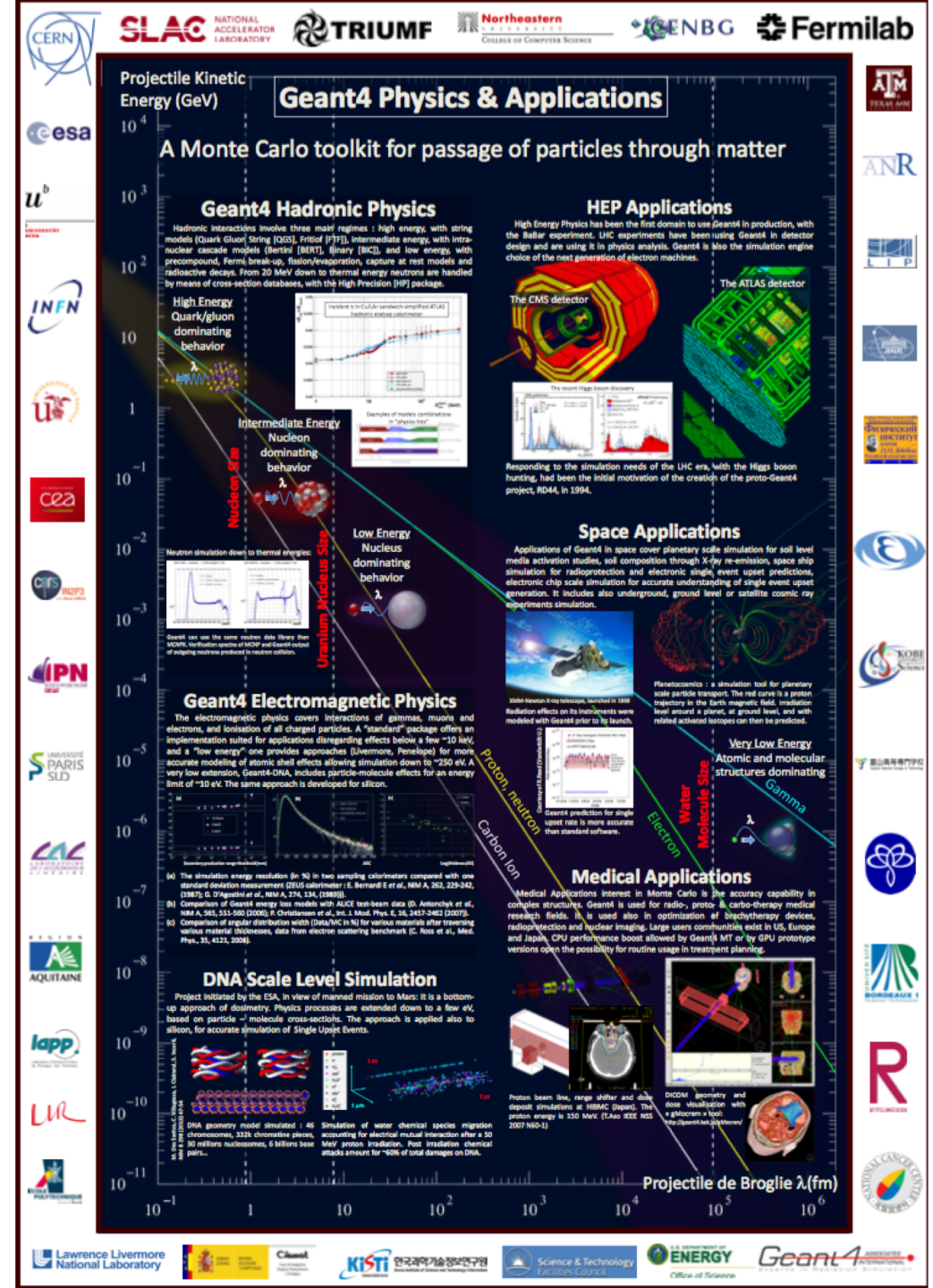
- Geant4 için ilk fikirler 1994 yılında San Francisco çalıştayında ortaya atılmıştır.
 - “Geant steps into the future” *R. Brun et al.*
 - “Object oriented analysis and design of a GEANT based detector simulator” *K. Amako et al.*
- Aralık 94 - R&D çalışmalara başlandı.
- Nisan 97 - İlk alpha dağıtımı
- Temmuz 98 - İlk beta dağıtımı
- Aralık 98- İlk Geant4 resmi yayını ver:1.0
- Neredeyse her sene yeni bir dağıtım çıkmaktadır. (Örn: v.10.1, v.10.2...)
- Mart 2015 - Geant4 v:10.0 dağıtımı
- Mart 2022 - Geant4 v:11 dağıtımı



Güncel
versiyon!

Geant4 Geçmişi ve Kullanım Alanları

- Hadronik Fizik
- YEF Uygulamaları
- Elektromagnetik Fizik Uygulamaları
- Uzay Çalışmaları / Uydu Teknolojileri
- DNA ölçekli Genetik Çalışmalar
- Sağlık Fiziği Uygulamaları (SPECT, PET, CT)

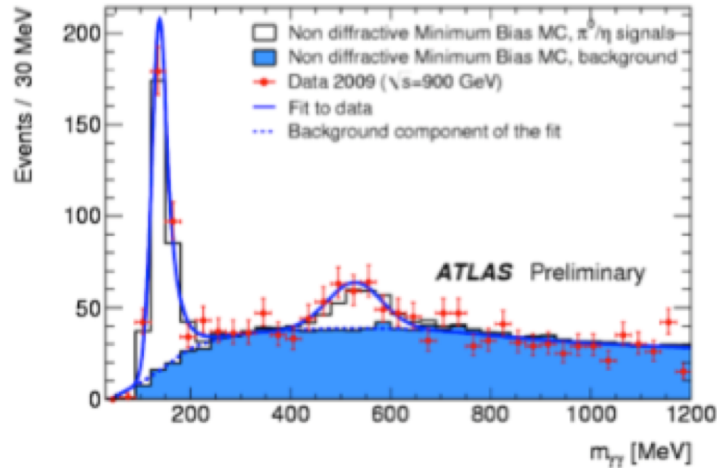


ATLAS ve CMS Deneyleri

- Detector design
- Calibration / alignment
- First analyses

T. LeCompte (ANL)

GEANT4 Comparisons with the Calorimeters



Invariant mass of pairs of well-isolated electromagnetic clusters.

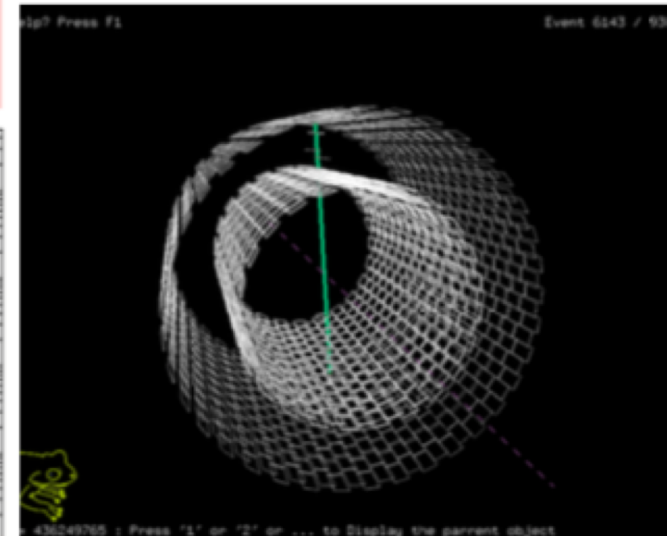
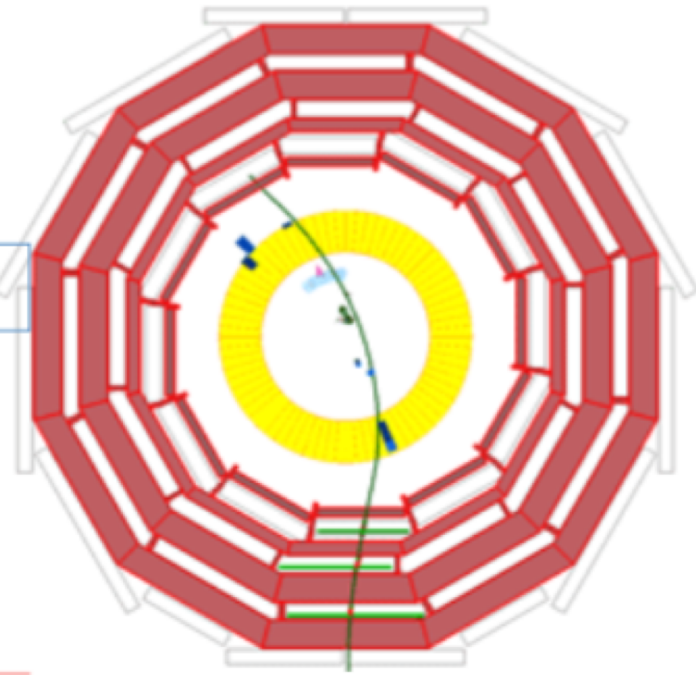
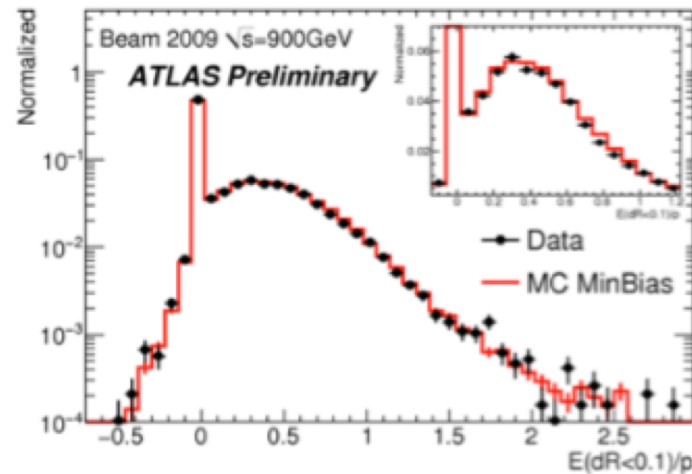
The π^0 mass is within $0.8 \pm 0.6\%$ of expectations.

The η^0 mass is within $3 \pm 2\%$ of expectations.

The detector uniformity is better than 2%.

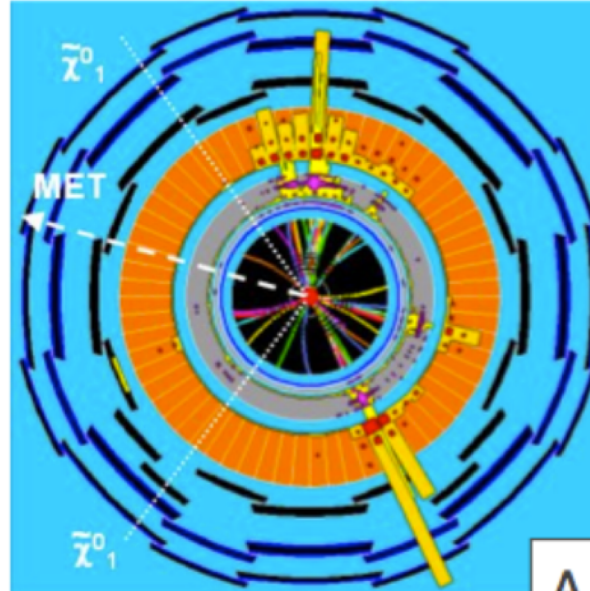
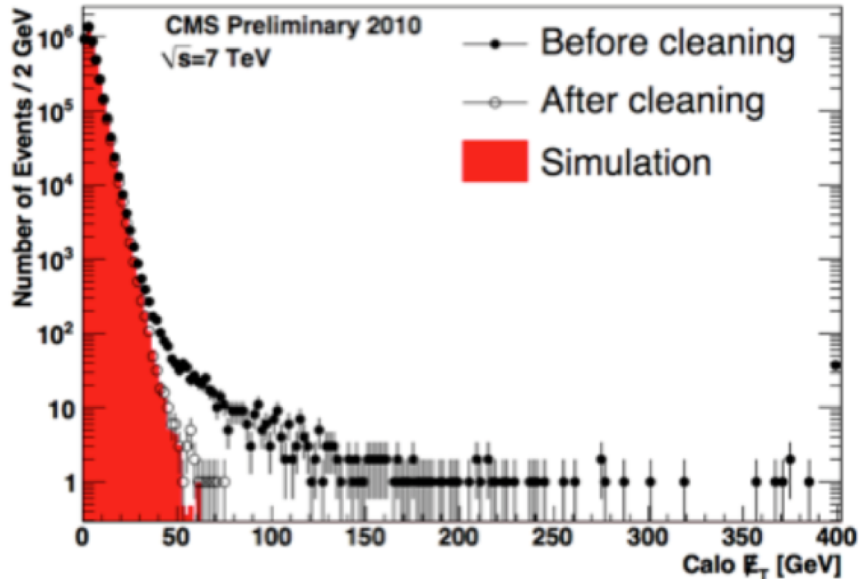
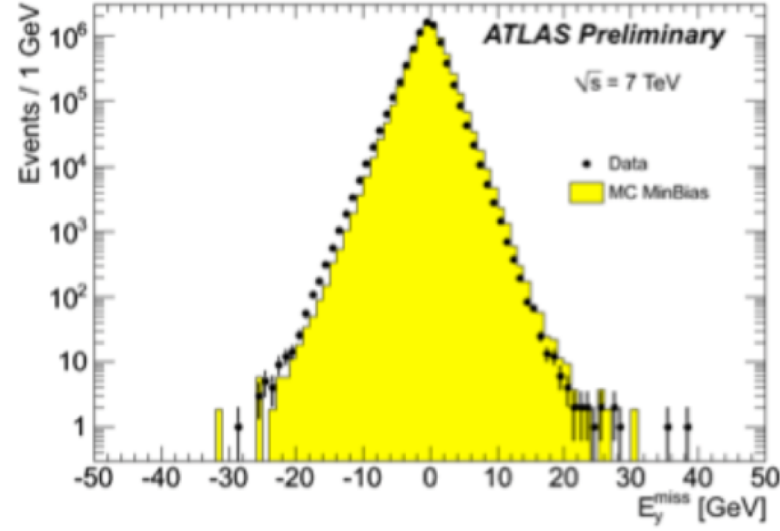
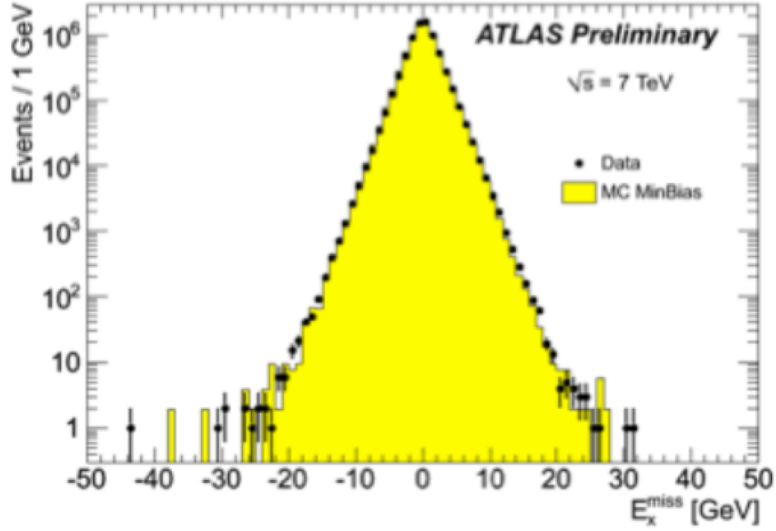
Response of the calorimeter to single isolated tracks. To reduce the effect of noise, topological clusters are used in summing the energy.

This plot agreed better than we ever expected. (I sent the student who made it back to make sure that they didn't accidentally compare G4 with G4.



Figures from CMS

ATLAS ve CMS Deneyleri

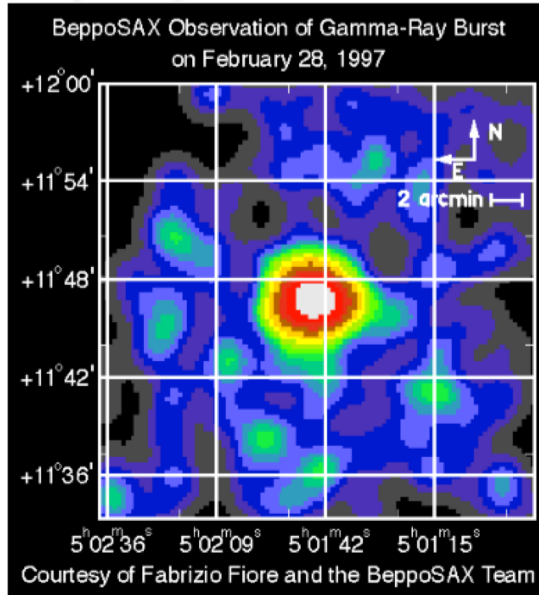


A GEANT4 event.

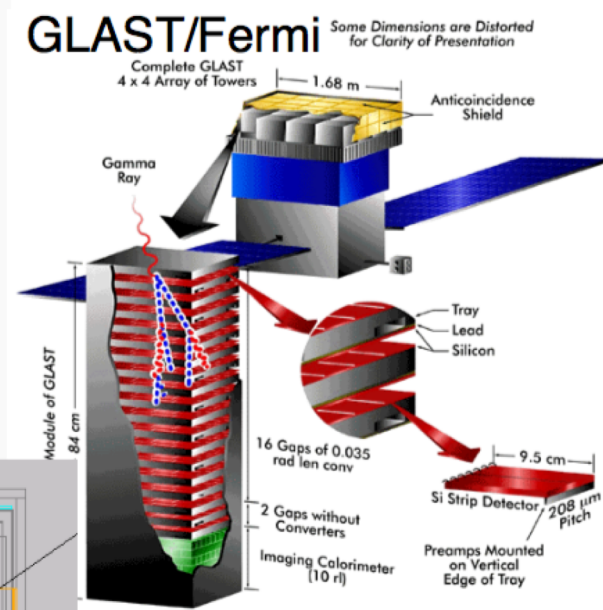
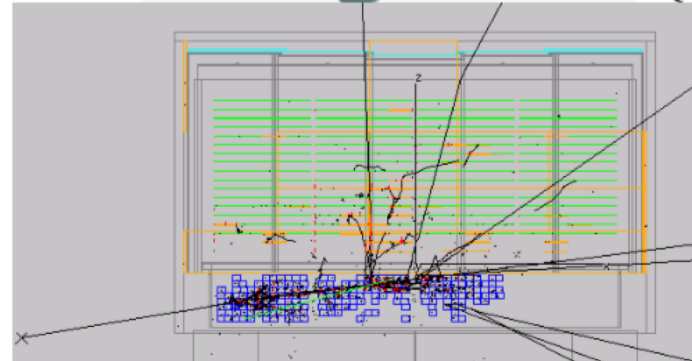
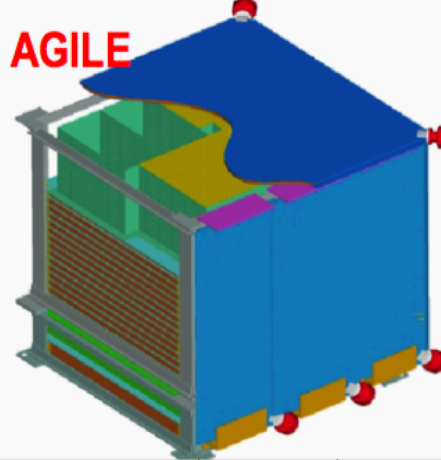
Elde edilmesi **güç** olan dağılımlardan biri **Kayıp Enine Enerjidir**. Detektör ile etkileşmeyen nötrino, karanlık madde yada bilinmeyen parçacıkların enerjilerinin benzetimi yapılmıştır. Deneysel uyum mükemmele yakındır.

γ astrophysics

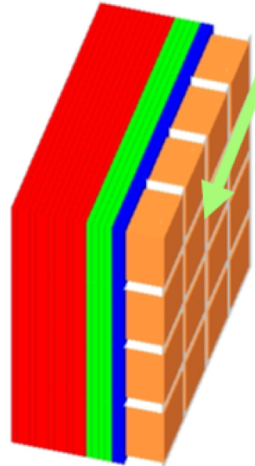
γ -ray bursts



AGILE

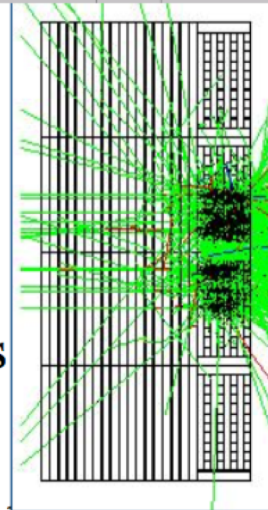


GLAST / Fermi



Typical telescope:
Tracker
Calorimeter
Anticoincidence

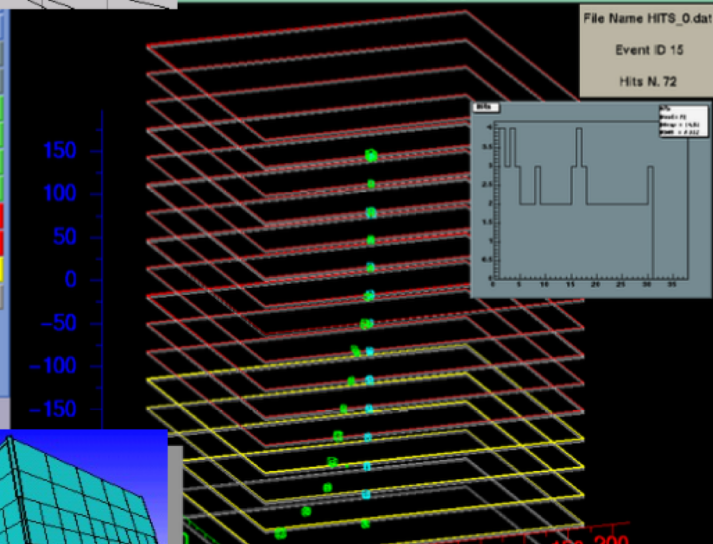
- γ conversion
- electron interactions
- multiple scattering
- δ -ray production
- charged particle tracking



- Previous
- View XZ
- View YZ
- Zoom
- Unzoom
- New Center
- Reset 3D
- Marker +
- Marker -
- Save as Gif
- View X3D



GLAST Hits Display



Geant 4

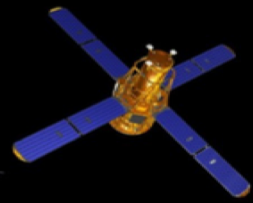
SLAC

Kernel I - M.

Geant4 in space



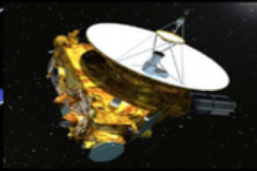
Akebono



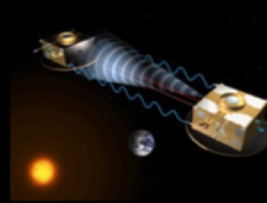
RHESSI



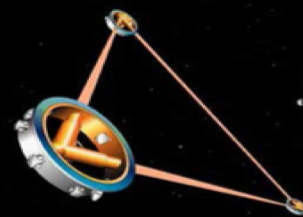
ACE



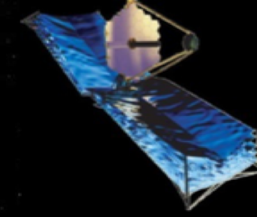
New Horizons



LISA Pathfinder



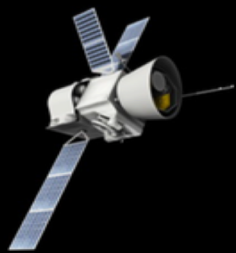
LISA



JWST

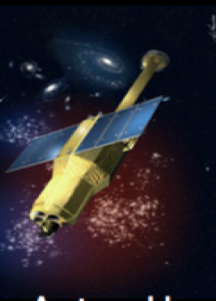
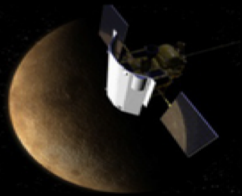


INTEGRAL



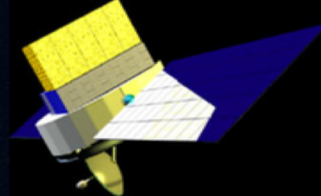
BepiColombo

Messenger



Astro-H

Fermi

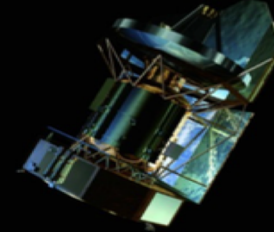


SOHO

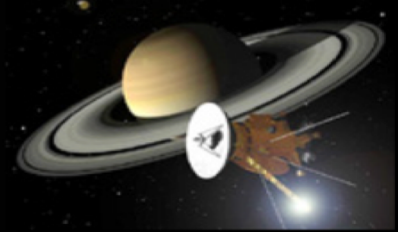


GAIA

Herschel



Cassini



Suzaku



SWIFT



XMM-Newton

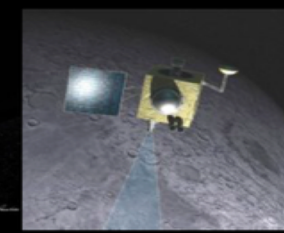
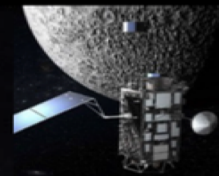


JUICE

JUNO



Kaguya



Chandrayaan-1



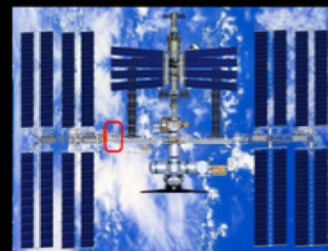
Chandrayaan-2



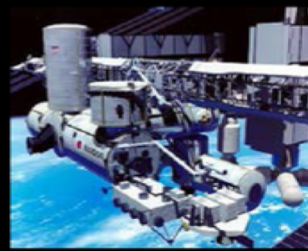
Columbus



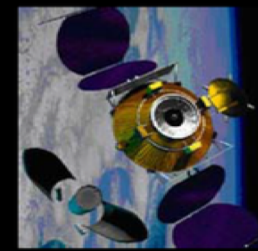
EUSO



AMS



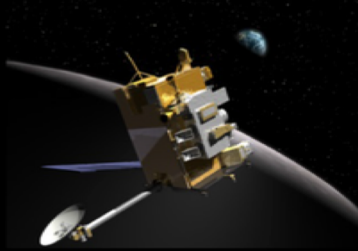
MAXI



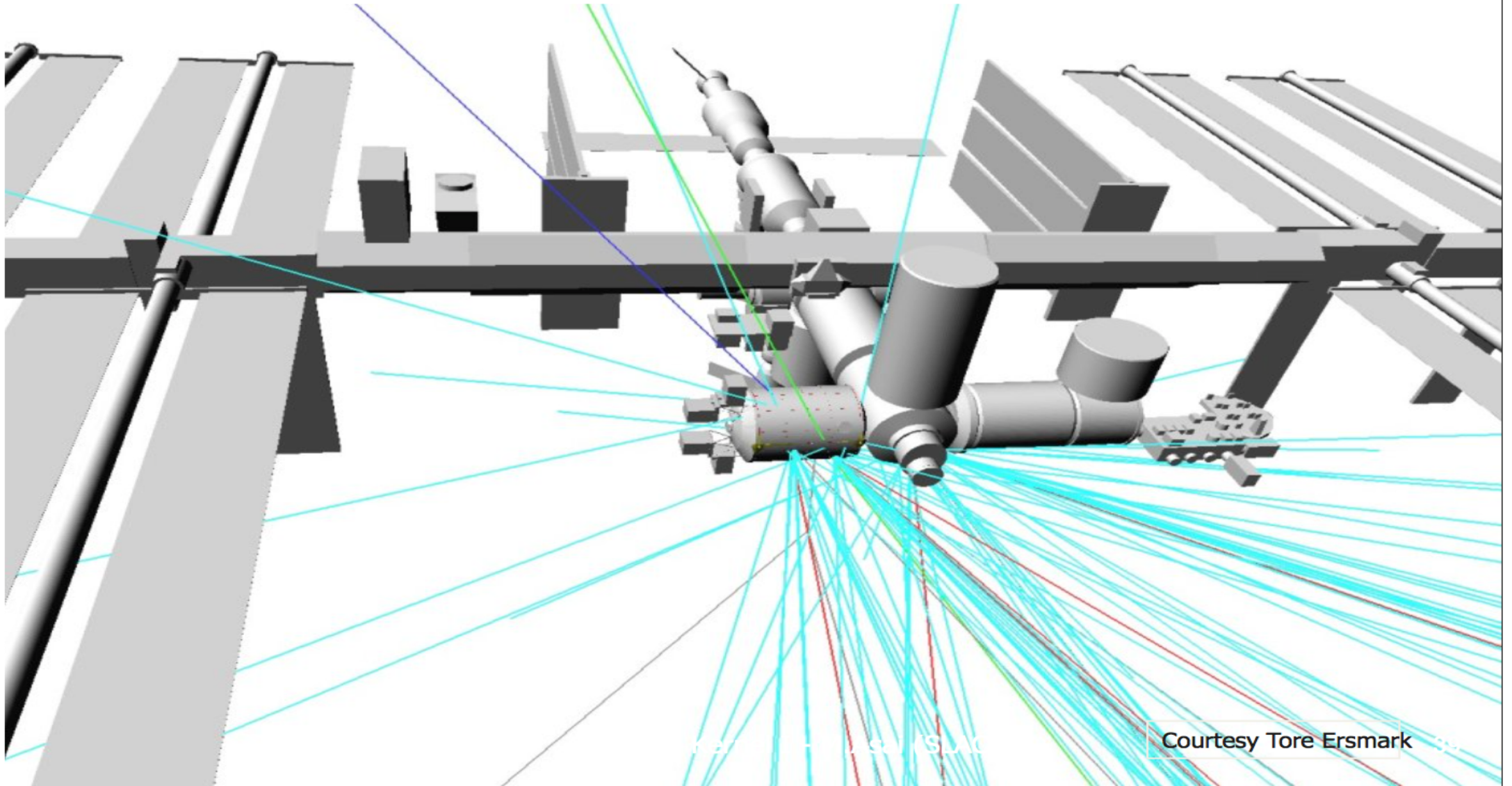
ConeXpress



Chang'e-1



LRO





PlanetoCosmics

Geant4 simulation of Cosmic Rays in planetary Atmo-/Magneto- spheres

SLAC

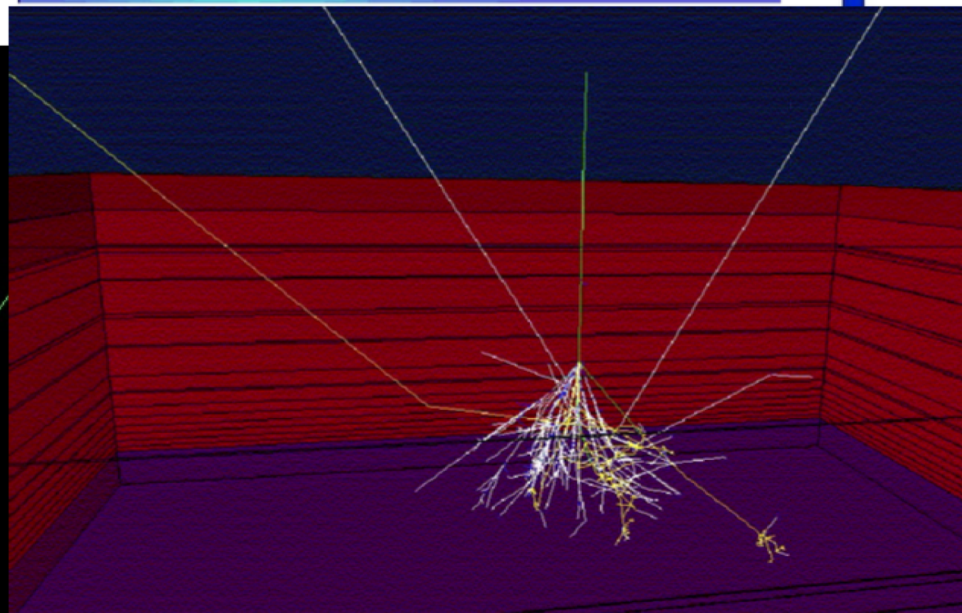
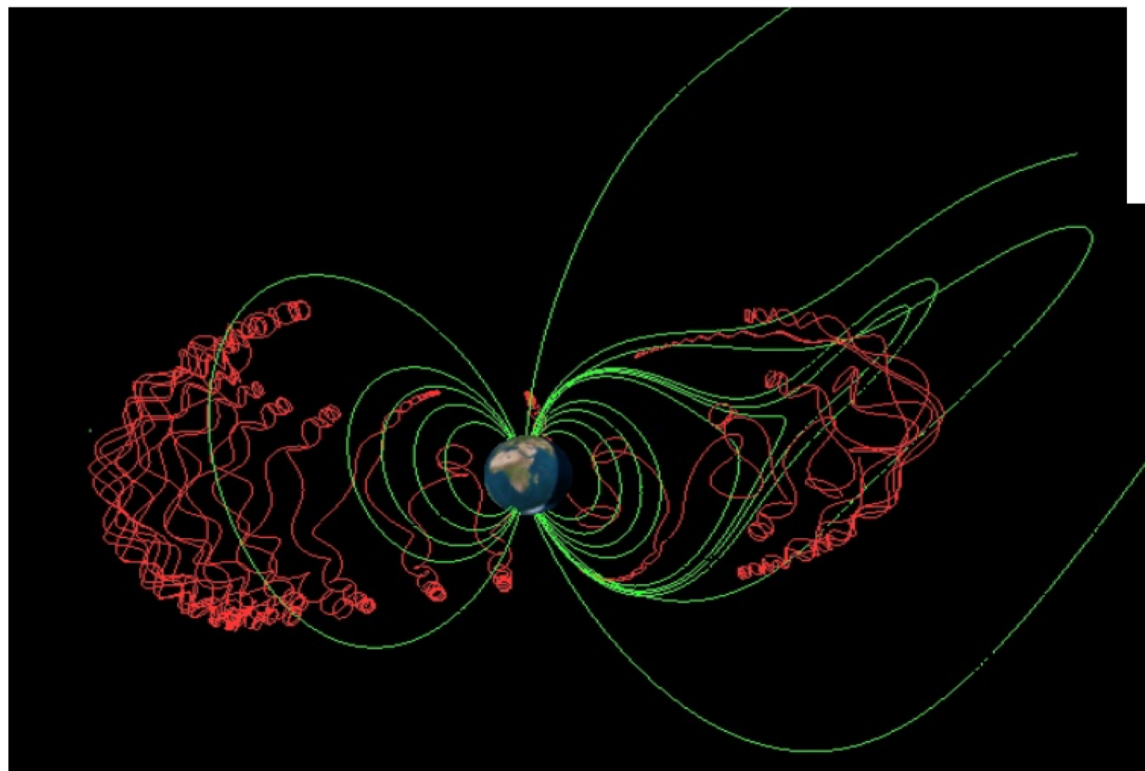
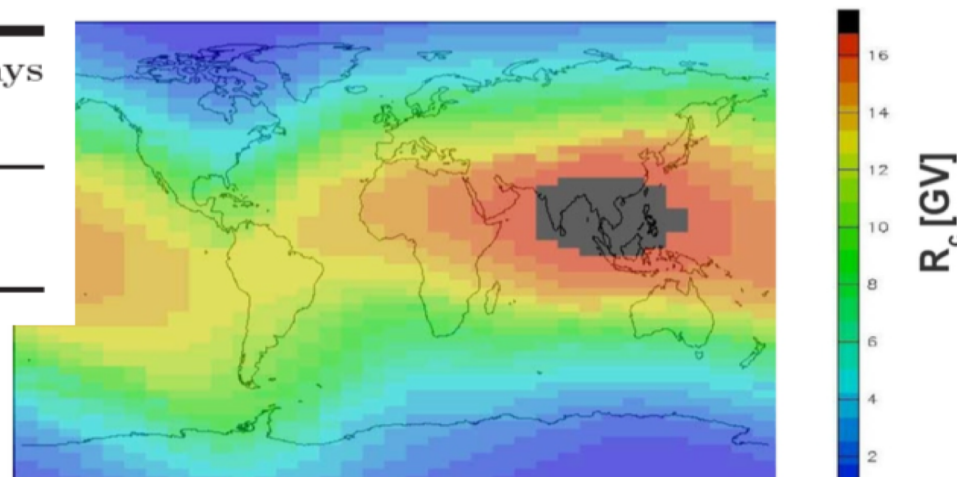
28th International Cosmic Ray Conference

— 4277

Cutoff Rigidities vs position

Geant4 Simulation of the Propagation of Cosmic Rays
through the Earth's Atmosphere

L. Desorgher, E. O. Flückiger, M. R. Moser, and R. Bütikofer
Physikalisches Institut, University of Bern, CH-3012 Bern, Switzerland

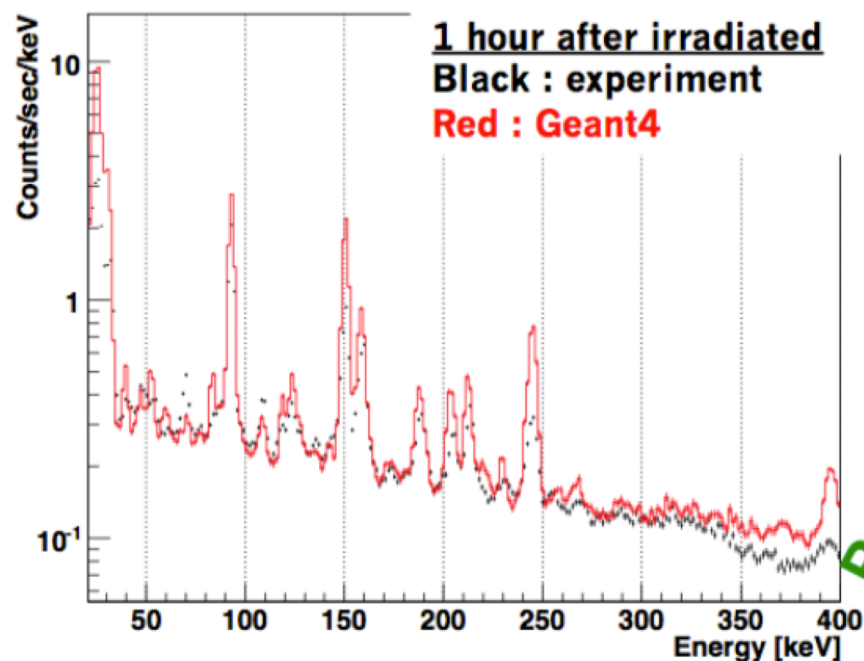




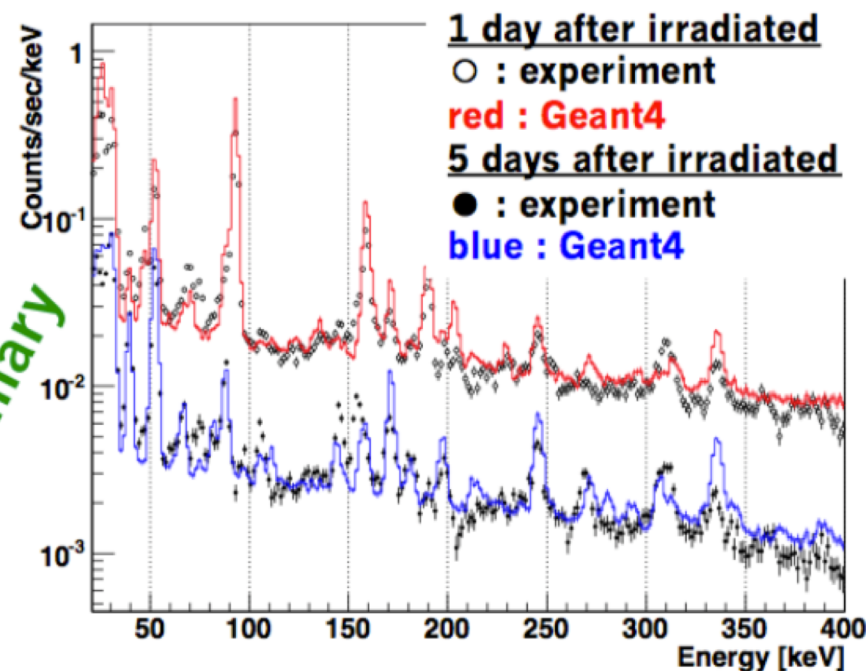
Time evolution of the activation background



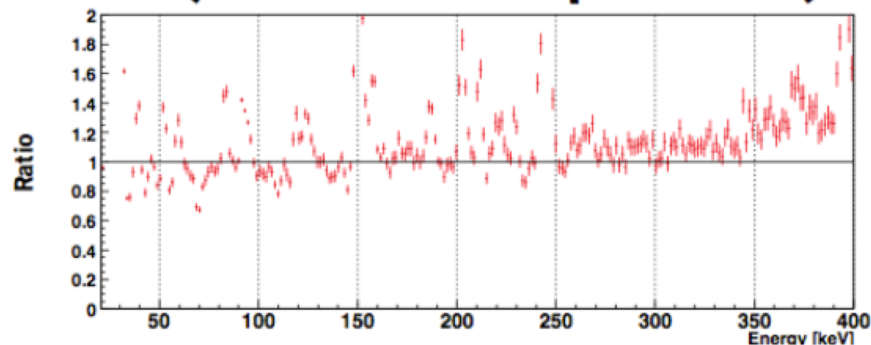
Comparison with Geant4



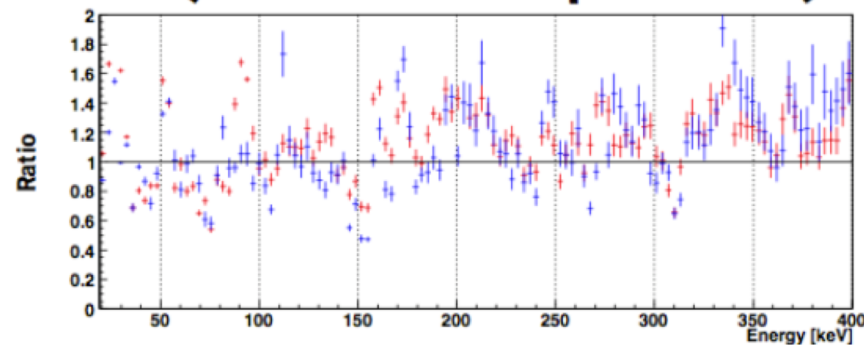
Preliminary



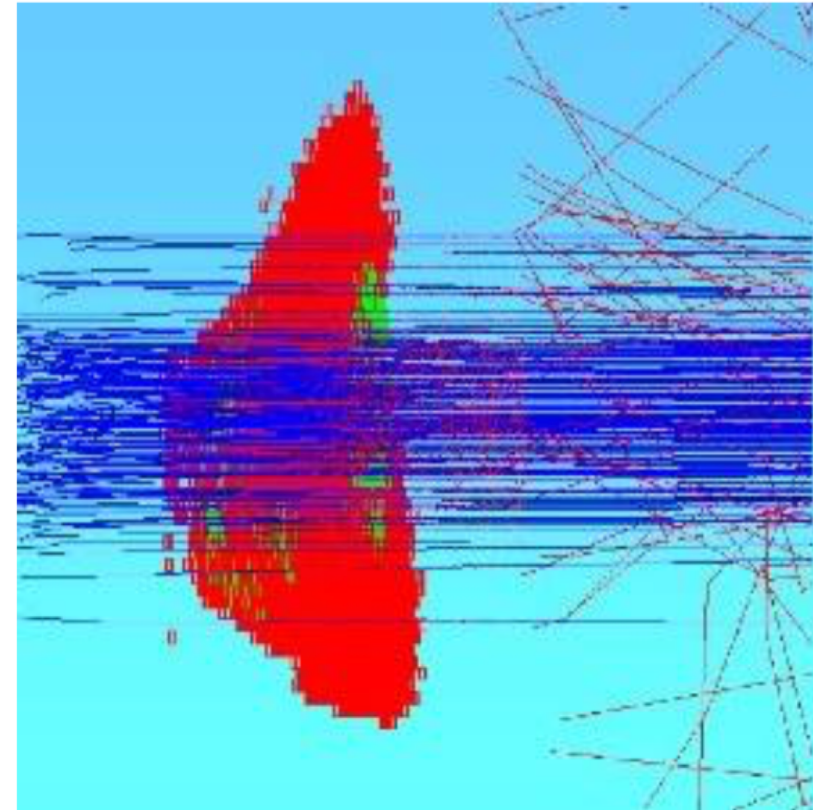
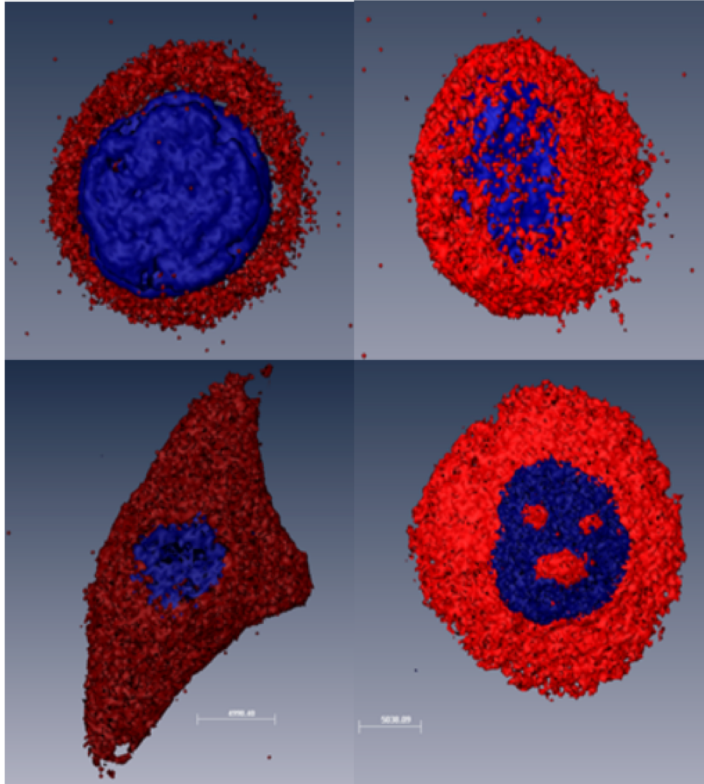
Ratio (simulation/experiment)



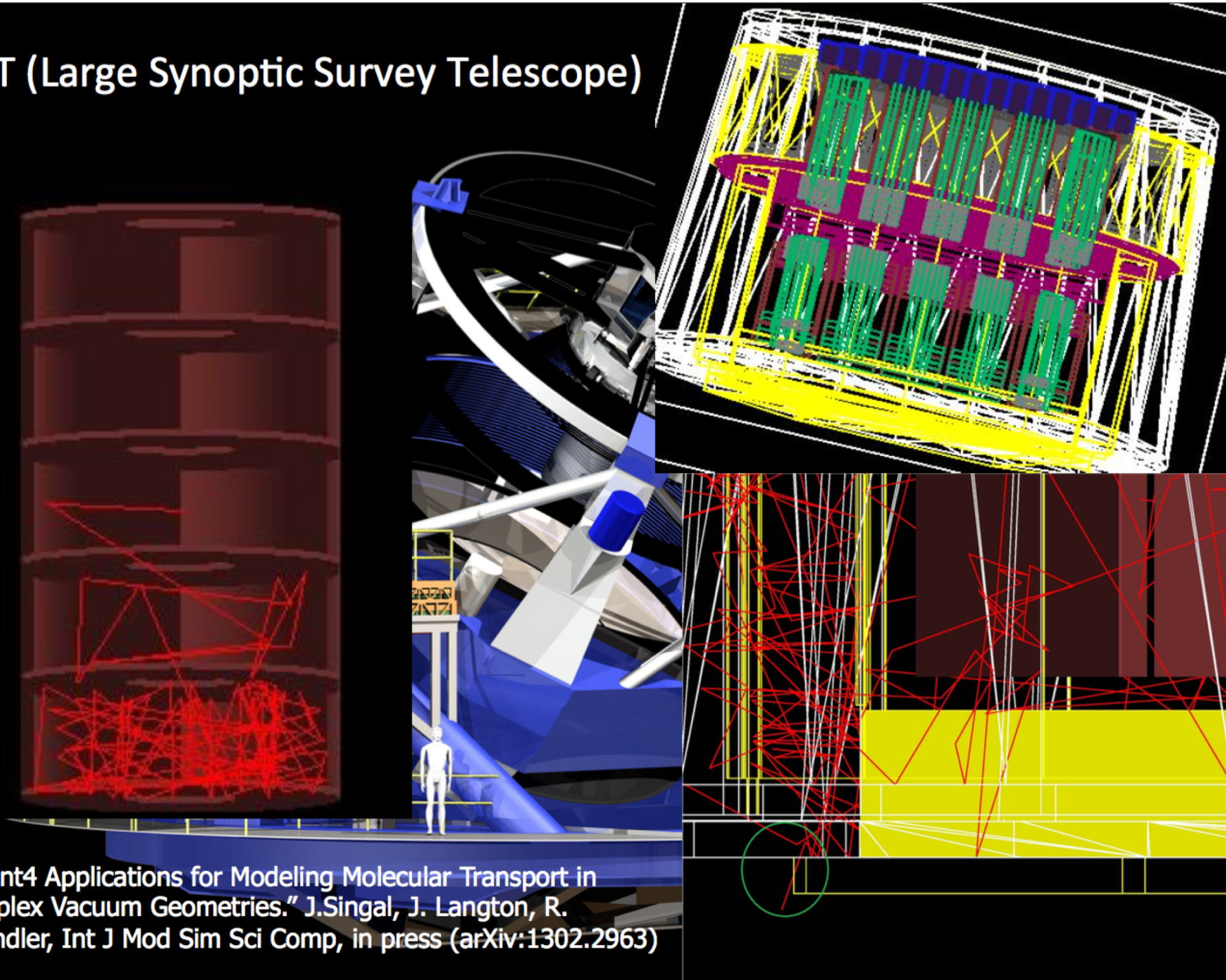
Ratio (simulation/experiment)



- 3 MeV enerjili alfa parçacıklarının yüksek çözünürlüklü hücresel fantom içindeki ışıması : “Single Cell Irradiation”
- $0.36 \times 0.36 \times 0.16$ mikron³ voxel boyutu ve 64 x 64 x 60 px çözünürlük
- **Courtesy of Sebasien Incerti (IN2P3-CNRS / CENBG)**



LSST (Large Synoptic Survey Telescope)



"Geant4 Applications for Modeling Molecular Transport in Complex Vacuum Geometries." J.Singal, J. Langton, R. Schindler, Int J Mod Sim Sci Comp, in press (arXiv:1302.2963)

Kısım 2:

Geant4 Kurulumu ve Terminolojisi

Geant4 Kurulumu

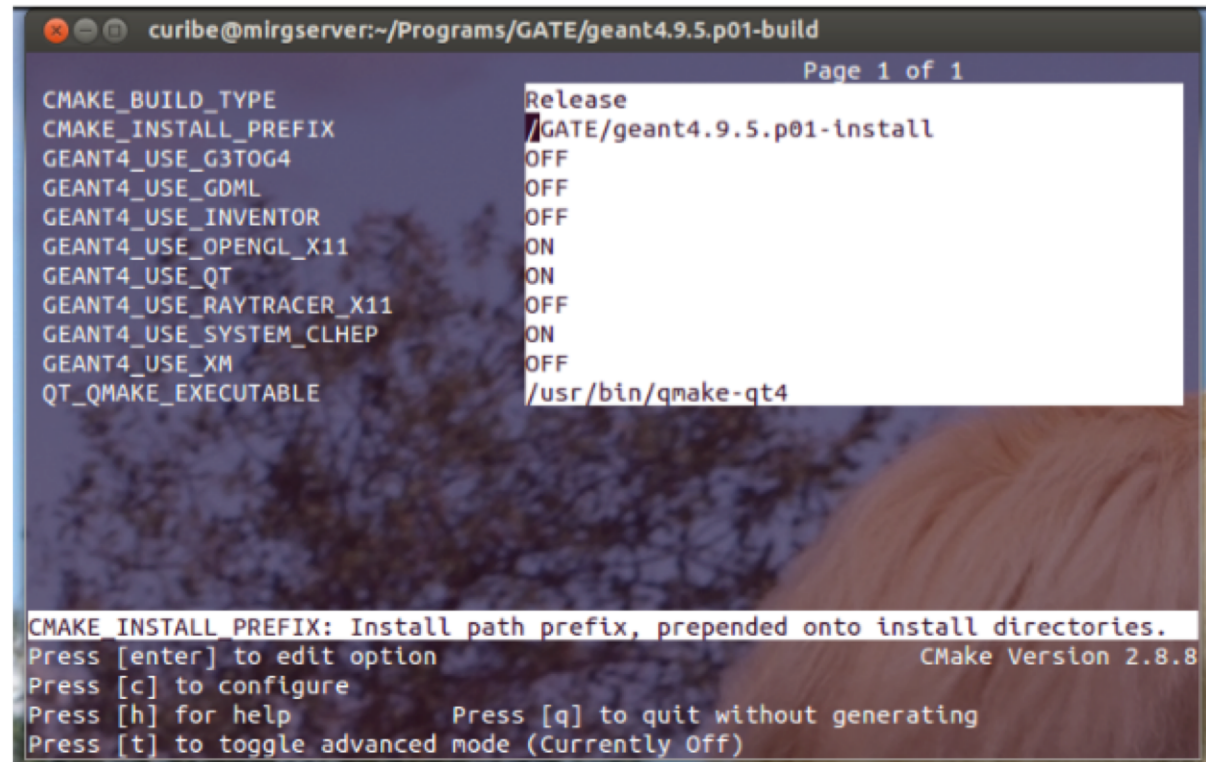
- **Kaynaktan kurulumu sistem yapısına göre 30 dk. ile 2 saat arasında sürmektedir!!**
- İşletim Sistemi: Linux / Unix, Windows (Visual Studio 20XX)
- Minimum Gereksinimler: CMake v > 3.3, Gcc [c++] v > 4 , CLHEP: 2.4.1.0, Expat 2.0.1 ve görselleştirme sürücülerі*
- Görselleştirme Sürücüler: Aşağıdakilerden biri yeterlidir: Qt4, OpenGL, MesaGL, X11/XQuartz, Motif UI
- Ayrıntılar:
<https://geant4-userdoc.web.cern.ch/UsersGuides/InstallationGuide/html/>

Bazı paket yöneticiler ile hızlı kurulum için:
Spack, HomeBrew, Conda, Macports, CVMFS

Geant4 Kurulumu

```
$ mkdir geant4.10.05-build
$ cd /path/to/geant4.10.05-build
$ cmake -DCMAKE_INSTALL_PREFIX=/path/to/geant4.10.05-install /path/to/geant4.10.05
veya
( $ cmake -DCMAKE_INSTALL_PREFIX=/path/to/geant4.10.05-install /path/to/geant4.10.05 )
$ make install
```

CCMake Kurulum Ekranı



GEANT4 ile Çalışmak

G4 erişimini
mümkün kılar

```
$ source geant4make.sh
```

```
G4ABLAData="/Users/kuday/geant4/build/data/G4ABLA3.0"  
G4ENSDFSTATEDATA="/Users/kuday/geant4/build/data/G4ENSDFSTATE1.2.3"  
G4INCLUDE="/Users/kuday/geant4/geant4.10.02.p02/this_is_a_deliberate_dummy_path"  
G4INSTALL="/Users/kuday/geant4/geant4.10.02.p02"  
G4LEData="/Users/kuday/geant4/build/data/G4EMLOW6.48"  
G4LEVELGAMMADATA="/Users/kuday/geant4/build/data/PhotonEvaporation3.2"  
G4LIB="/Users/kuday/geant4/build/BuildProducts/lib"  
G4LIB_BUILD_SHARED="1"  
G4LIB_USE_EXPAT="1"  
G4LIB_USE_ZLIB="1"  
G4NEUTRONHPDATA="/Users/kuday/geant4/build/data/G4NDL4.5"  
G4NEUTRONXSData="/Users/kuday/geant4/build/data/G4NEUTRONXS1.4"  
G4PIIData="/Users/kuday/geant4/build/data/G4PII1.3"  
G4RADIOACTIVEDATA="/Users/kuday/geant4/build/data/RadioactiveDecay4.3.2"  
G4REALSURFACEDATA="/Users/kuday/geant4/build/data/RealSurface1.0"  
G4SAIDXSDATA="/Users/kuday/geant4/build/data/G4SAIDDATA1.1"  
G4SYSTEM="Darwin-clang"  
G4UI_USE_TCSH="1"  
G4VIS_USE_OPENGLX="1"  
G4WORKDIR="/Users/kuday/geant4_workdir"
```

Not: 'Export' komutuyla oturum boyunca tanımlı tüm çevresel değişkenleri kontrol edebilirsiniz.

Eğer G4.. değişkenleri tanımlı değilse geant4 kurulumu başarı ile gerçekleşmemiştir !!

Tanımlı olması gereken
değişkenler çevresel tanımlanır.

**vGate8.2 için geant4 değişkenlerinin tanımlanması
aşağıdaki komutla sağlanır:**

```
source /usr/share/geant4/geant4.10.03.p03-build/geant4make.sh
```

*source geant4make.sh komutu .bashrc / .profile gibi çekirdek konfigürasyon betiklerine eklenerek açılışta çalışması sağlanabilir.

İlk Örneğin Derlenmesi

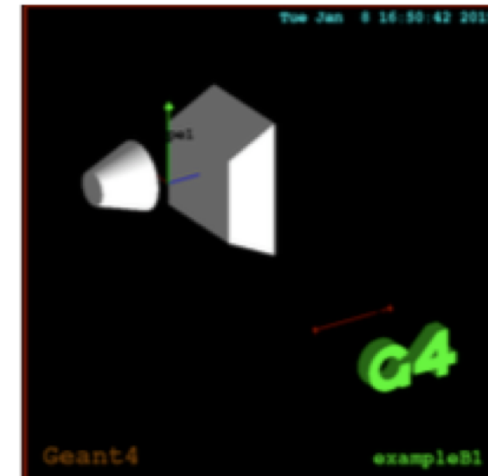
```
$ cp -r geant4/geant4.10.02.p02/examples/basic/B1 .  
$ cd B1  
$ make
```

Örneği kaynak
dosyasından dizine
kopyalar.

```
Making dependency for file exampleB1.cc ...  
Making dependency for file src/B1SteppingAction.cc ...  
Making dependency for file src/B1RunAction.cc ...  
Making dependency for file src/B1PrimaryGeneratorAction.cc ...  
Making dependency for file src/B1EventAction.cc ...  
Making dependency for file src/B1DetectorConstruction.cc ...  
Making dependency for file src/B1ActionInitialization.cc ...  
Compiling B1ActionInitialization.cc ...  
Compiling B1DetectorConstruction.cc ...  
Compiling B1EventAction.cc ...  
Compiling B1PrimaryGeneratorAction.cc ...  
Compiling B1RunAction.cc ...  
Compiling B1SteppingAction.cc ...  
Creating shared library /Users/kuday/geant4_workdir/tmp/Darwin-clang/exampleB1/libexampleB1.dylib  
...  
Compiling exampleB1.cc ...  
Using global libraries ...  
Linking exampleB1  
... Done!
```

GNUmakefile
dosyası çalışır.

```
$ exampleB1  
veya ($ ./exampleB1)
```



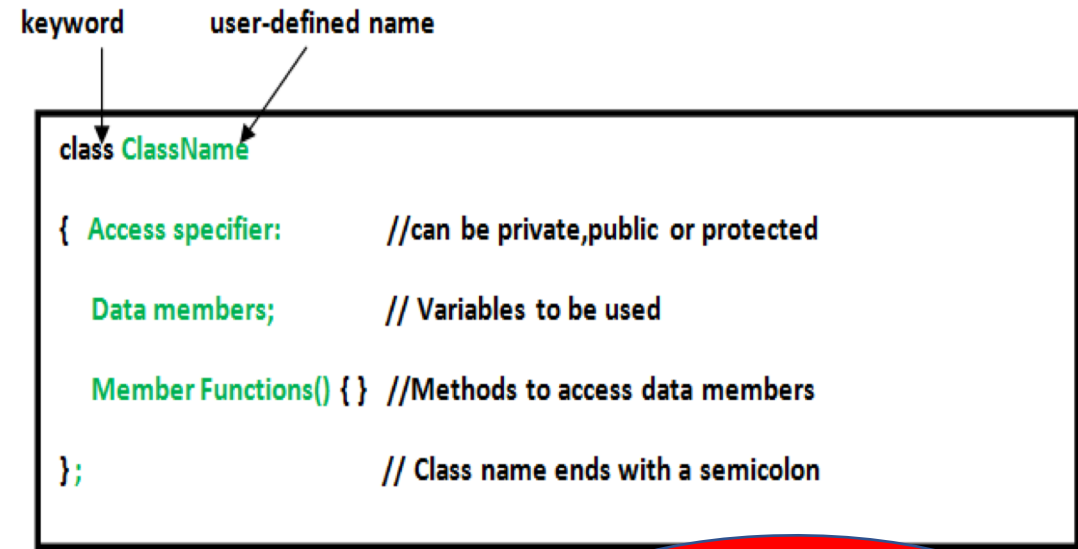
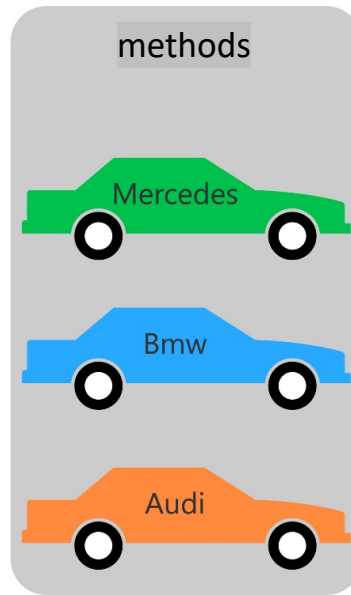
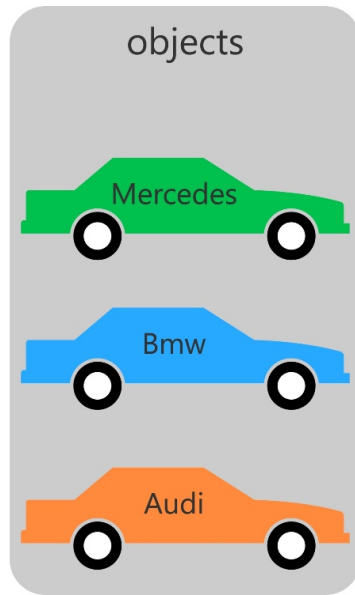
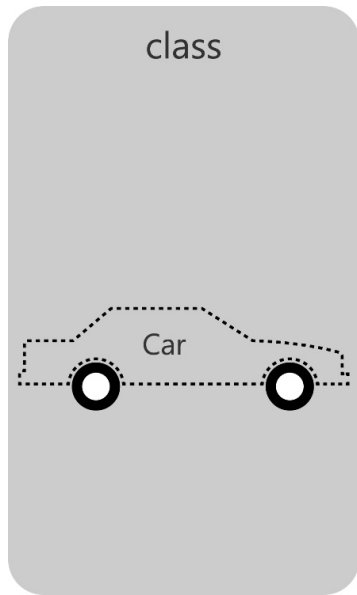
Geant4 Hızlı Kurulumu

Daha hızlı bir kurulum Geant4 tabanlı çalışan bazı uygulama paketlerinin sanal sistemde (Virtual Machine) kurulumu ile gerçekleştirilebilir. (Eğitim amaçlı)

ÖRNEKLER:

1. **GATE -> vGATE** : www.opengatecollaboration.com
2. **GAMOS** : <https://sites.dartmouth.edu/optmed/research-projects/monte-carlo-software/>
3. **Geant4 VM**: <https://extra.lp2ib.in2p3.fr/G4/>

C++ Obje Yönelimli Programlama (Sınıf Yapısı)



```
class CCar  
{  
private:  
    int m_iNumberOfBolts;  
public:  
    int m_iCarCurrentSpeed;  
    CCar()  
    {  
        m_iCarCurrentSpeed = 0;  
        m_iCurrentCarGear = 0;  
        m_iNumberOfBolts = 78569;  
    }  
protected:  
    int m_iCurrentCarGear;  
    int GetCurrentSpeed() {return m_iCarCurrentSpeed;}  
};
```

CCar Mercedes;
CCar Bmw;
CCar Audi;

<Sınıf Adı> Obje Adı;

Mercedes.GetCurrentSpeed();
Bmw.GetCurrentSpeed();
Audi.GetCurrentSpeed();

Bmw.OpenBaggageDoor();

«0»

ERROR: Undefined method in class !!!

Tüm Geant4 Kütüphanesi bu tür sınıflardan oluşur

Örnek G4 main.cc
kodları bu
yapıdadır!!

C++ Obje Yönelimli Programlama (Türetilmiş Sınıf Yapısı)

Temel sınıf sentaksı:

```
#include<iostream>
using namespace std;
class base {
public:
base()
{
cout<<"base class constructor"<<endl;
}
~base()
{
cout<<"base class destructor"<<endl;
}
};
```

Temel sınıflar G4 kütüphanesinde zaten yaratılmıştır. Kod içinde `#include base.hh` şeklinde çağrılırlar.

Türetilmiş sınıf sentaksı:

```
class derived:public base
{
public:
derived()
{
cout<<"derived class constructor"<<endl;
}
~derived()
{
cout<<"derived class destructor"<<endl;
}
};
int main()
{
derived d;
return 0;
}
```

G4 örneklerinin tamamı ve bizim yaratacağımız kodlarda src ve inc içindeki tüm kodlar bu yapıdadır!!

Dersin geri kalan kısmında G4 kütüphanesinde yer alan önemli (zorunlu) sınıflar tanıtılacaktır...

Geant4 Terminolojisi

- Run, event, track, step, step point
- Track (İz) <-> Trajectory (yörünge), step <-> yörünge noktaları
- Process (At rest, along step, post step)
- Cut = production threshold (Üretim Eşik Değerleri)
- Sensitive detector (SD) , score, hit, hits collection.
- Multithreaded

Normalde tüm izler, sıfır kinetik enerjiye kadar izlenir.

SD: Detektörün read-out parçalarının benzetimidir. (Sınıf ismi)

Kodu, çok işlemcili sistemlerde çalıştırmak için bölümlere ayırmak (threaded)

Geant4 Terminolojisi (RUN)

- Run: Çalışma (Koşma)
- Gerçek deneylerde; deney sisteminin yeterince ölçüm alıncaya kadar çalıştırılarak işletilmesi süresine bir çalışma (run) denir.
- Benzetimlerde de aynı terminoloji kullanılır.
- Geant4 için “**BeamOn**” komutu çalışmayı başlatan komuttur.
- Benzetimlerde **tek çalışma bir olay döngüsünün gerçekleştiği aralıktır.**
- **G4RunManager** sınıfı çalışmanın yönetimini ve temsilini sağlar.
- **G4Run** sınıfı kullanıcı tanımlı çalışmaların türetilmesini sağlar. (Bir çalışma objesi o çalışmadaki özet bilgileri içerebilir.)
- **G4RunUserAction** opsiyonel olarak kullanılabilir.

Geant4 Terminolojisi (EVENT)

- Event: Olay - G4 benzetiminin en küçük birimidir.
- Benzetim başladığında öncül (primer) izler birleştirilerek istiflenir.
- İstiflenen izler tek tek ele alınarak ikincil (sekonder) izler belirlenir.
- Elde edilen tüm izlerin istiflenmesi bitinceye kadar işlem devam eder.
- Böylece “Tracking” işlemi yapılmış olur.
- **G4Event** sınıfı tek bir olayı temsil eder. İşlem sonunda aşağıdaki nesnelere sahip olur
 -  Input: Öncül köşeler (primary vertices) ve parçacık listesi
 -  Output: Hit sayıları ve Yörüngeler (Trajectories)
- **G4EventManager** sınıfı bir olayın üretilmesini yönetir. Opsiyonel olarak kullanıcı tanımlanmış olaylar **G4UserEventAction** ile yönetilir.

Bir olay, detektörde birçok Hit'e karşılık gelebilir !!

Geant4 Terminolojisi (TRACK)

- Track: İz - G4 içinde bir parçacığın anlık (ekran) görüntüsüdür.
- Parçacığın o an içindeki kısa zaman aralığındaki fiziksel miktarıdır.
- Step tanımıyla karıştırılmamalıdır.
- İzler hızlıca işlemlenir ve kinetik enerjisi sıfıra düşünce silinir.
- Geometrilerin dışına çıkan, bozunmaya ve çarpışmaya uğrayan izler de silinebilir.
- **G4Track** sınıfı tek bir izi temsil eder. Bir olayın bitiminde hiçbir iz nesnesi devam edemez.



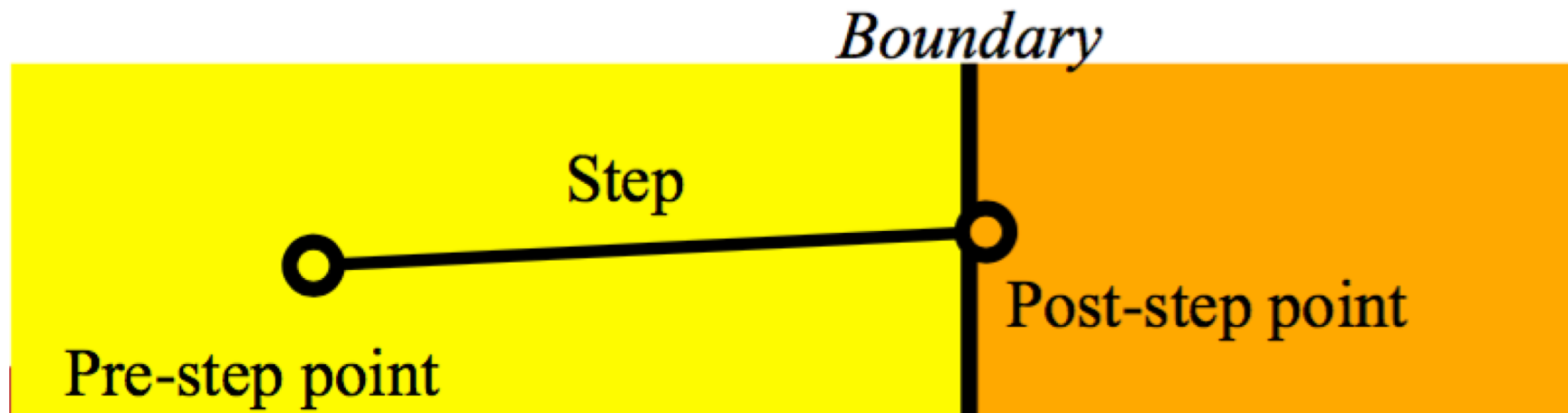
Tüm izlerin kaydedilmesi ile yörünge (trajectory) sınıfı oluşur.

- **G4TrackingManager** sınıfı bir izin üretilmesini yönetir. Opsiyonel olarak kullanıcı tanımlı izler **G4UserTrackingAction** ile yönetilir.

Geant4 Terminolojisi (STEP)

Enerji kaybı ve ToF

- Step: Adım - G4 içinde bir parçacığın delta bilgisini taşır.
- Adım uçlarındaki noktalar boyunca hacim ve materyal bilgisi tutulur.
- Eğer noktalardan birisi sınıra denk geldiyse; sonraki hacime ait olarak kaydedilir.
- **G4Step** sınıfı tek bir adımı temsil eder.
- **G4SteppingManager** sınıfı bir adımın üretilmesini yönetir. Opsiyonel olarak kullanıcı tanımlı adımlar **G4UserSteppingAction** ile yönetilir.



Geant4 Terminolojisi (TRAJECTORY)

- Trajectory: Yörünge - G4 iz bilgisinin kaybolmadan depolandığı sınıftır.
- Adım uçlarındaki noktalar boyunca hacim ve materyal bilgisi tutulur.
- **G4Trajectory** sınıfı tek yörünge oluşumunu temsil eder.
 -  G4Track nesnesinin bazı bilgilerini kopyalar.
- **G4TrajectoryPoint** sınıfı yörünge noktasının oluşumunu temsil eder.
 -  G4Step nesnesinin bazı bilgilerini kopyalar.
- G4Trajectory nesnesi G4TrajectoryPoint sınıfından bir vektöre sahiptir.
- Event işleme sonucunda, G4Events nesnesi bir dizi G4Trajectory nesneleriyle dolar (dolmalıdır) (**//tracking/storeTrajectory must be set to 1**)
- Programların çok fazla miktarda yörünge depolamamasına dikkat edilmelidir. Hafıza ve işleme yavaşlayabilir !!

Geant4 Terminolojisi (PARTICLE)

- Particle: Parçacık - G4 içinde 4 farklı sınıf yoluyla tanımlanır:
- **G4Track** sınıfı;
 - ➡ Pozisyon, geometrik yapı .vs.
 - ➡ Bu sınıf izi alınacak parçacığa ilişkilendirilmiş olmalıdır.
- **G4DynamicParticle** sınıfı;
 - ➡ Momentum, enerji, spin gibi parçacığın dinamik özellikleri saklanır.
 - ➡ Her G4 Track nesnesinin kendi G4DynamicParticle nesnesi vardır.
 - ➡ Tekil (bireysel) olarak parçacığın temsil edildiği sınıftır.
- **G4ParticleDefinition** sınıfı;
 - ➡ Yük, kütle, bozunum kanalı ...vs. gibi statik bilgiler saklar.
 - ➡ G4ProcessManager nesnesine ait G4ParticleDefinition nesnesi var.
 - ➡ G4ParticleDefinition tip1 <—> G4DynamicParticle dtip1, dtip2..vs.

Geant4 Terminolojisi (SD)

- Verilen geometride, fizik etkileşimleri olup izler üretilirken ve olaylar ortaya çıkarken detektöre ulaşan sadece hit (entry) bilgisi vardır.
- Kullanıcılar tüm süreçten bilgi alabilmek için 3 yola başvurabilir;
 - Gömülü skarlama komutlarını kullanabilir.
 - İzlenen hacim için skarlama kullanabilir.
 - Detektör read-out (sensörü) ile hitlerin sayımı yapılmalıdır.
 - ★ Bu yöntem için **G4VSensitiveDetector** nesnesi bir hacme atanmalıdır.
- Kullanıcı tanımlı (kendi oluşturduğu) sınıf veya yöntemleri de kullanabilir.
 - Bknz. **G4VUserTrackingAction**, **G4VUserSteppingAction**

Geant4 Terminolojisi (UNITS)

- Dahili birim sistemleri G4 kodu içinde saklı olarak görünür. Fakat kısaltmalar aşıkardır (trivial)
- Birimlendirilmek istenen her sayı ilgili **birim ismiyle çarpılır**.

`radius = 10.0 * cm;`

`Mass = 100.0 * GeV;`

- Dönüşümü istenen birimler için bölme işlemi yapılır.

`G4cout << eDep / MeV << " [MeV]" << G4endl;`

- Kullanıcı arayüzünde `/units/list` komutu ile varsayılan birimler görüntülenebilir.
- Birimler belirtilmezse varsayılan birimlerde çıktı elde edilir.

Geant4 Terminolojisi (COUT / CIN)

- G4 tarafında tanımlı *iostream* nesneleri **G4cout** ve **G4cin** komutları kullanılarak dışa / içe aktarılır.



G4cerr ve **G4endl** de ihtiyaca göre kullanılabilir.

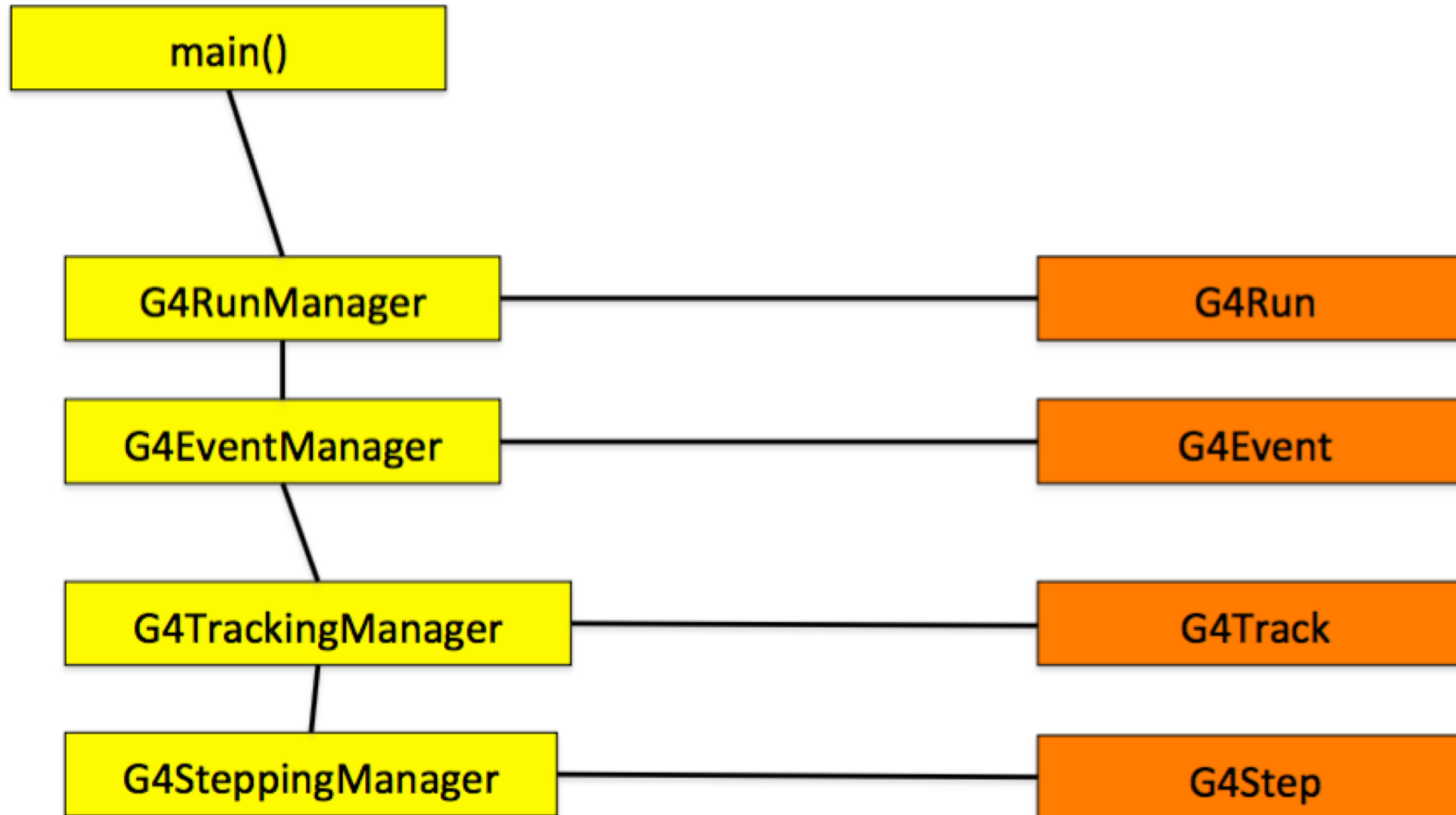
Örn:

```
G4cout << "Merhaba Bodrum!!" << Gendl;
```

- Kullanıcıların `std::cout` / `std::cin` kullanması G4 kodu içinde **uygun değildir !!!**
- Dosya işlemleri için I/O bilinen C++ komutları geçerlidir.
- **G4cout** ve **G4cin** komutları özellikleri kaydedilen değerlerin takip edilmesi ve kodun doğrulanması açısından önemli olabilir!!

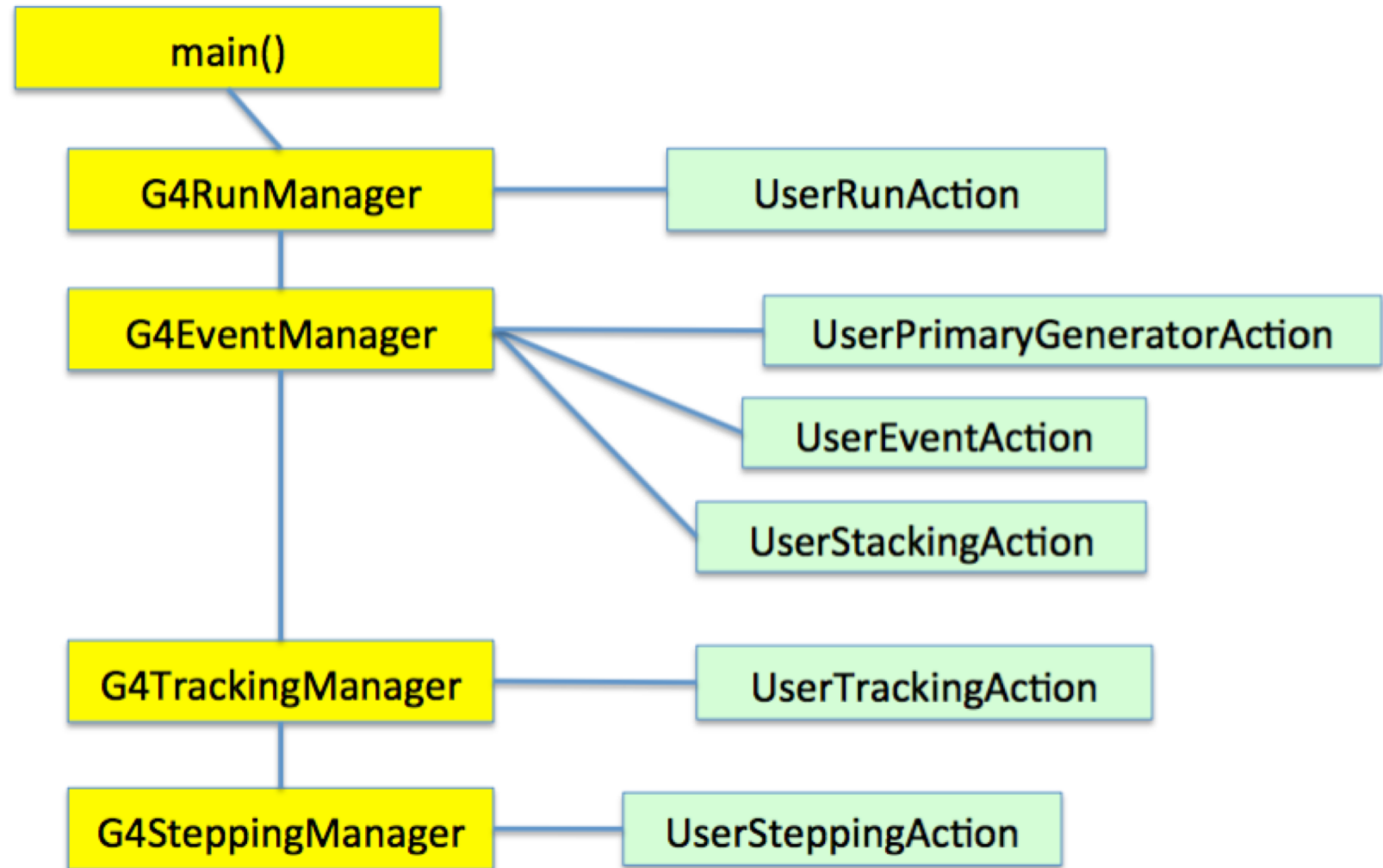
Geant4 Terminolojisi (PROCESS)

- main() çalışması sequantial mod için;



Geant4 Terminolojisi (PROCESS)

- main() çalışması sequantial mod için kullanıcı tanımlı eylemler (action) ile;



Geant4 Terminolojisi (main.cc kodu nasıl olmalı?)

- Tipik bir G4 uygulamasında; int main() içinde ;
 - ▶ Geometrik kurulumu tanımlayın;
 - Materyal ve Hacimler
 - ▶ Fiziksel kurulumu tanımlayın;
 - Parçacıklar, fiziksel etkileşimler, modeller ...vs.
 - Üretim eşik değerleri
 - ▶ Olayı ve olay döngüsünü tanımlayın;
 - Öncül izlerin (primary tracks) üretimi
 - Olaylardaki size gerekli bilgileri kaydedin
 - ▶ Çıktılarını uygun formatta yazdırın
 - ▶ Görselleştirme işlemleri (opsiyonel)

SonuMessenger (Örn:G4DetectorMessenger) ile biten tüm sınıflar ve kod dosyaları, makro kartlarında yer alacak komutların tanımlandığı kodlardır.

Yani macro komutları ile C++ komutları arasındaki bağlantı ...Messenger kodlarında tanımlanır.

Kısım 3:

Kullanıcı Arayüzü & Temel

Komutlar

- Bir örneğin derlendikten sonra makrosuz çalıştırılmasıyla kullanıcı arayüzü açılır. Eğer gerekli şekilde ayarlandıysa pencere şeklinde de açılabilir.

```
*****
Geant4 version Name: geant4-10-02-patch-02    (17-June-2016)
Copyright : Geant4 Collaboration
Reference : NIM A 506 (2003), 250-303
WWW : http://cern.ch/geant4
*****
```

```
Visualization Manager initialising...
Registering graphics systems...
```

```
You have successfully registered the following graphics systems.
Current available graphics systems are:
```

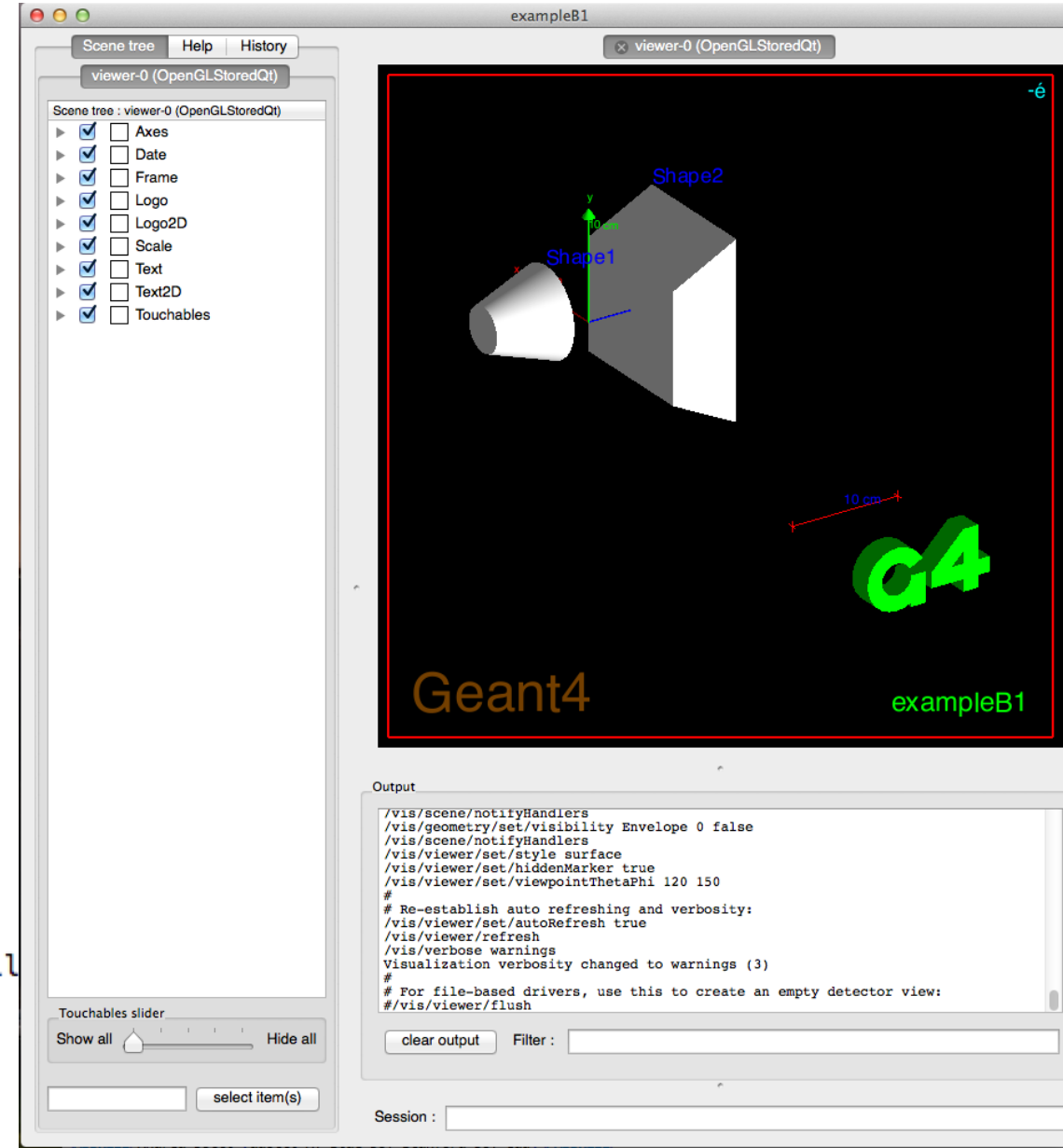
```
*****
*****
*****
```

```
You have successfully registered the following user vis actions.
Run Duration User Vis Actions: none
End of Event User Vis Actions: none
End of Run User Vis Actions: none
```

```
Some /vis commands (optionally) take a string to specify colour.
Available colours:
    black, blue, brown, cyan, gray, green, grey, magenta, red, white, yell
```

```
Available UI session types: [ GAG, tcsh, csh ]
PreInit>
```

Kullanıcı Arayüzü (UI)



Kullanıcı Arayüzü (UI) - Komut Sentaksı

- UI komut sentaksı 3 argümanlı yazılabilir;
 - **Komut dizini** `/run/verbose 1`
 - **Komut** `/vis/viewer/flush`
 - **Parameter (ler)**
- Geant4, UI komutlarını 3 farklı şekilde alıp işlemleyebilir;
 1. Interaktif Kullanıcı arayüzünden yazılıp <enter> ile gönderilerek (Bknz. Önceki sayfa)
 2. Makro dosyasına komut ekleyerek
 3. Kod içinde kullanarak;
Örn: **G4UImanager* UI = G4UImanager::GetUIpointer();**
UI->ApplyCommand("/run/verbose 1");
- Unix komutları geçerli ve kullanışlıdır: **ls, cd, pwd, history, <tab>, help, exit**

Kullanıcı Arayüzü (UI) - Temel Komutlar

- `/control/alias [isim] [değer]`
- `/control/loop[macroDosyası][değişken][başlangıçDeğeri][sonDeğer][adım]`
- `/control/foreach [macroDosyası] [değişken] ["değer listesi"]`
- `/control/if [koşul] [macroDosyası]`
- `/units/list`
- `/run/initialize`
- `/run/beamOn [run başlatır]`
- `/gun/particle [parçacıkAdı]`
- `/gun/energy [sayı] [birim]`
- `/vis/verbose [sayı]`
- `help [Uikomut]`

main() fonksiyonu örneği;

```
int main(int argc, char** argv) {  
    ...  
    if (argc != 1)  
    { // batch mode  
        G4String command = "/control/execute ";  
        G4String fileName = argv[1];  
        UImanager->ApplyCommand(command+fileName);  
    }  
    else  
    { // interactive mode : define UI session  
        G4UIExecutive* ui = new G4UIExecutive(argc, argv);  
        ui->SessionStart();  
        delete ui;  
    }  
}
```


Teşekkürler!

**Uygulama: Geant4 ile B1
örneğini çalıştırınız.**

vGate ile İlgili Yardım:

- Virtual Box uygulaması çalıştırılır,
- **Start | Başlat** ikonu ile konuk işletim sistemi vGate başlatılır.
- **vGate sistemi içinde Admin Kullanıcı: vgate , Şifre: virtual**
- **Ayrıntılı yardım dökümanları:**
<https://opengate.readthedocs.io>
- **Bir Gate uygulaması çalıştırmak için:**
\$> Gate <uygulama_makrosu>.mac
- **vGate ve Gate detaylı bilgi için:**
<http://opengatecollaboration.org/>