Experimental Techniques in Particle Physics

Geant4:
Summary

Andreas Nowack
nowack@physik.rwth-aachen.de
RWTH Aachen University

WS 2020/21

Quick Intro to Geant 4

Topics

- overview, documentation (2020-10-29)
- geometry:
  - introduction, solids, logical/physical volumes (2020-11-05)
  - Boolean operations (2020-11-12)
  - multiple placements, hierarchy of volumes (2020-11-19)
  - replicas (2020-11-26)
- materials (2020-12-03)
- particles, processes (2020-12-17)
- user actions (2021-01-07)
- particle sources (2021-01-14)
- sensitive detectors (2021-01-21)

- evaluation test (2020-12-10, 2021-01-07)

Topics

- overview, documentation (2020-10-29)
- geometry:
  - introduction, solids, logical/physical volumes (2020-11-05)
  - Boolean operations (2020-11-12)
  - multiple placements, hierarchy of volumes (2020-11-19)
  - replicas (2020-11-26)
- materials (2020-12-03)
- particles, processes (2020-12-17)
- user actions (2021-01-07)
- particle sources (2021-01-14)
- sensitive detectors (2021-01-21)

- evaluation test (2020-12-10, 2021-01-07)

Geant 4 Overview

Recap

What GEANT acronym sta ???

What is Geant4?

A toolkit for the Passage of Particles through Matter

- Monte Carlo
- All Particle
- Complex Geometry
- Motion
- Fields
- Modern Programming (C++)
  - Hence FLEXIBLE
- Open and Free

I'm **version 4**

Geant 4 Overview

Recap

GEANT4 does not simulate beam-beam interactions.
One probably has to combine it with some beam-beam event generators (pythia, sherpa…)

GEANT4 is not an executable program/file.
It is a collection of libraries organized in classes (C++ OOP)

GEANT4 is not an analysis program.
One probably has to combine it with other OOP analysis tools (ROOT)

User Documentation

Recap

- Documentation (link to page with pdf version below)
  - Introduction to Geant4
  - Installation Guide
  - Application Developers Guide
  - Toolkit Developers Guide
  - Physics Reference Manual
  - Physics List Guide
  - Code Cross Reference – LXR
  - Classes and Members Reference Guide – Doxygen

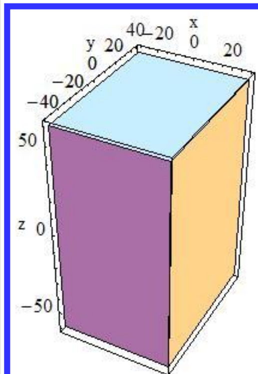
Topics

- overview, documentation (2020-10-29)
- geometry:
  - introduction, solids, logical/physical volumes (2020-11-05)
  - Boolean operations (2020-11-12)
  - multiple placements, hierarchy of volumes (2020-11-19)
  - replicas (2020-11-26)
- materials (2020-12-03)
- particles, processes (2020-12-17)
- user actions (2021-01-07)
- particle sources (2021-01-14)
- sensitive detectors (2021-01-21)

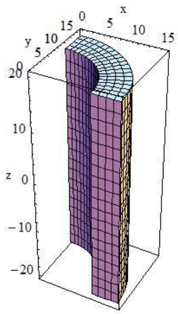
- evaluation test (2020-12-10, 2021-01-07)

How Many Predefined CSG Do We Have?

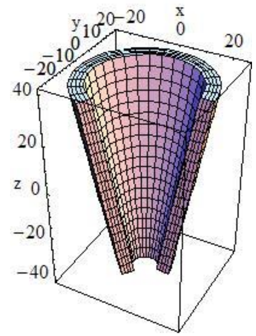
Recap



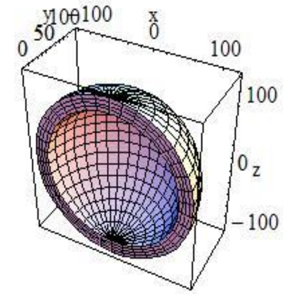
Box



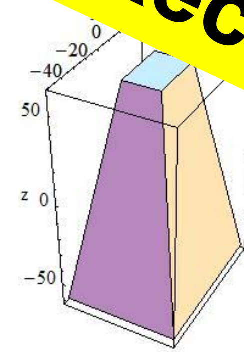
Tube



Cone

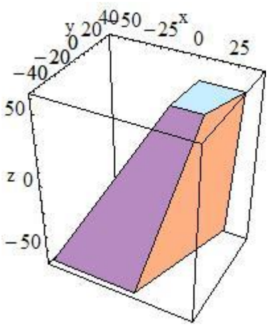


Sphere

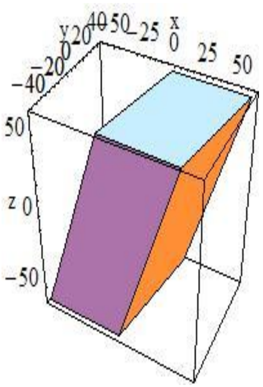


Trapezoid

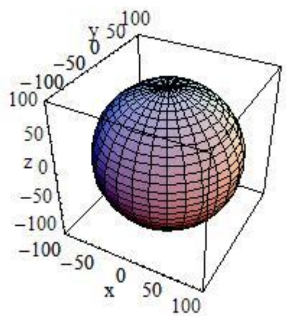
My preferred ones… very basic and simple… but very powerful



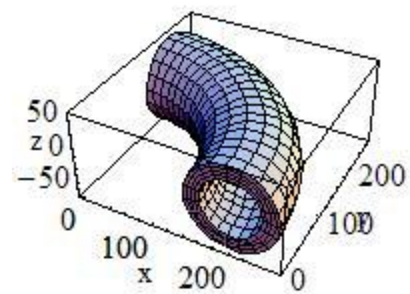
Generic Trapezoid



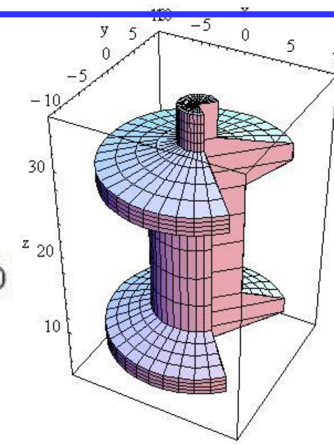
Parallelepiped



Solid Sphere



Torus



Polycons

Notion of World, Mother and Daughter Volumes

Recap

➡ **world volume is the mother volume of Volume 1 and 2**

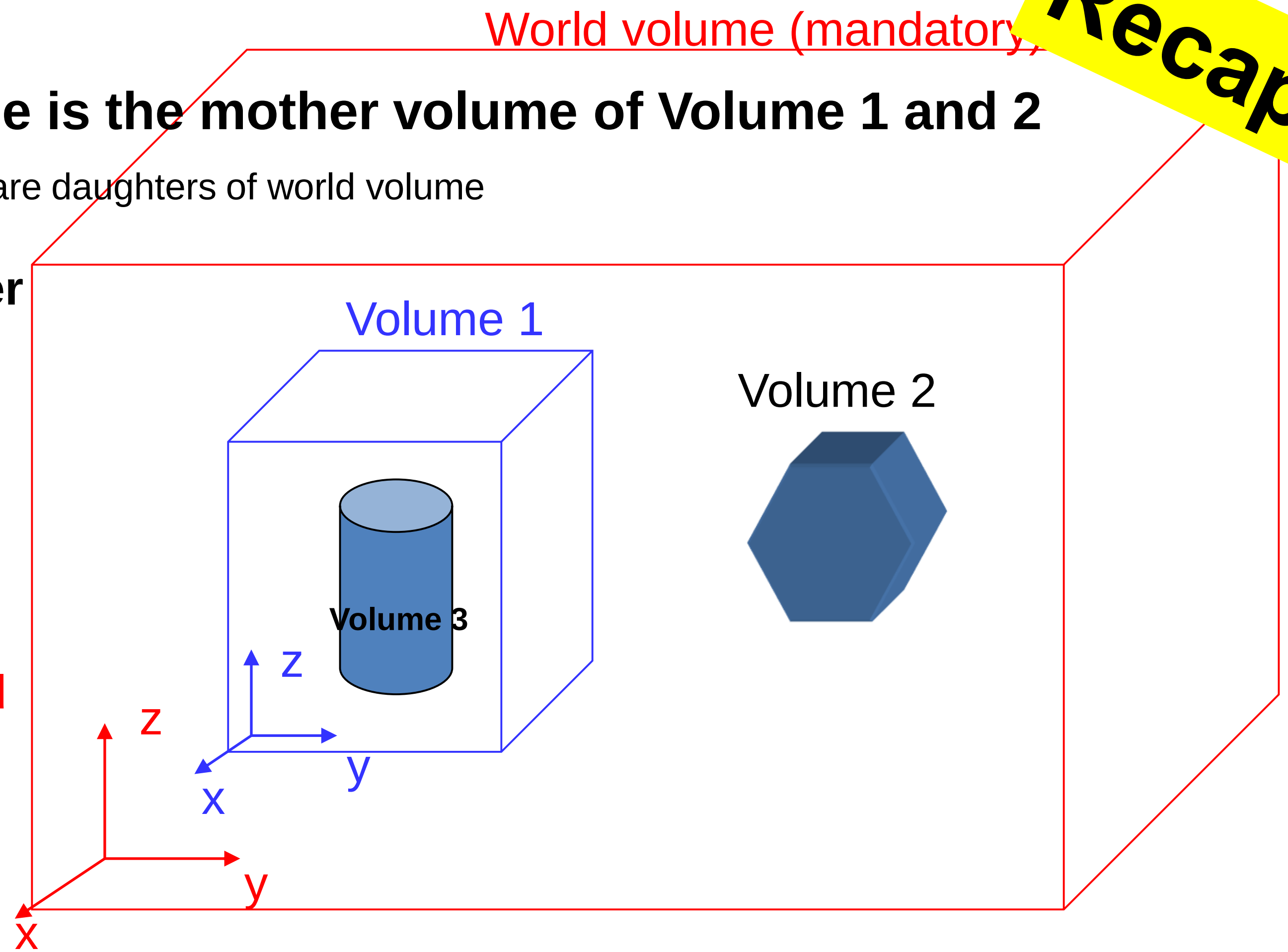
Volume 1 and 2 are daughters of world volume

Volume 1 is mother of volume 3

etc…

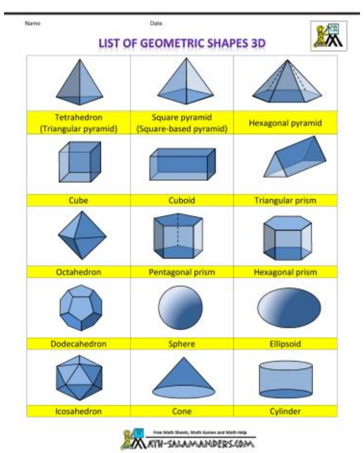
Important rules !

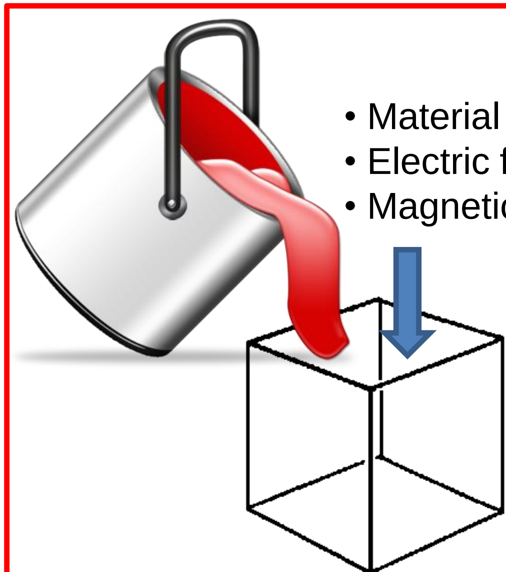
1. **No overlap**
2. **Fully contained**



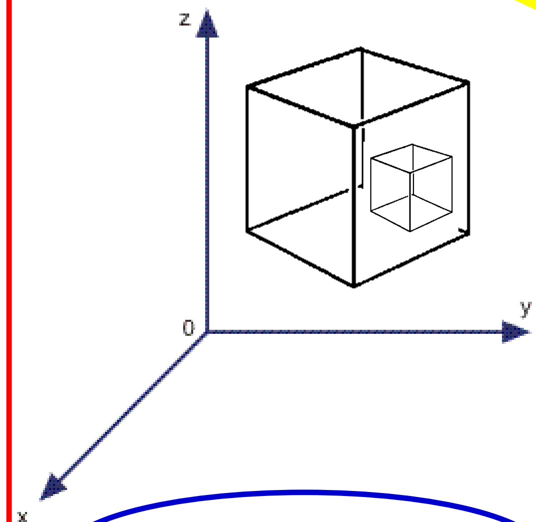
Geometry in Three Steps

Recap





- Material
- Electric field
- Magnetic field



Mother & Daughter!

Mathematical shape
(Solid)

Logical Volume

Placement in (X, Y, Z)
Physical Volume

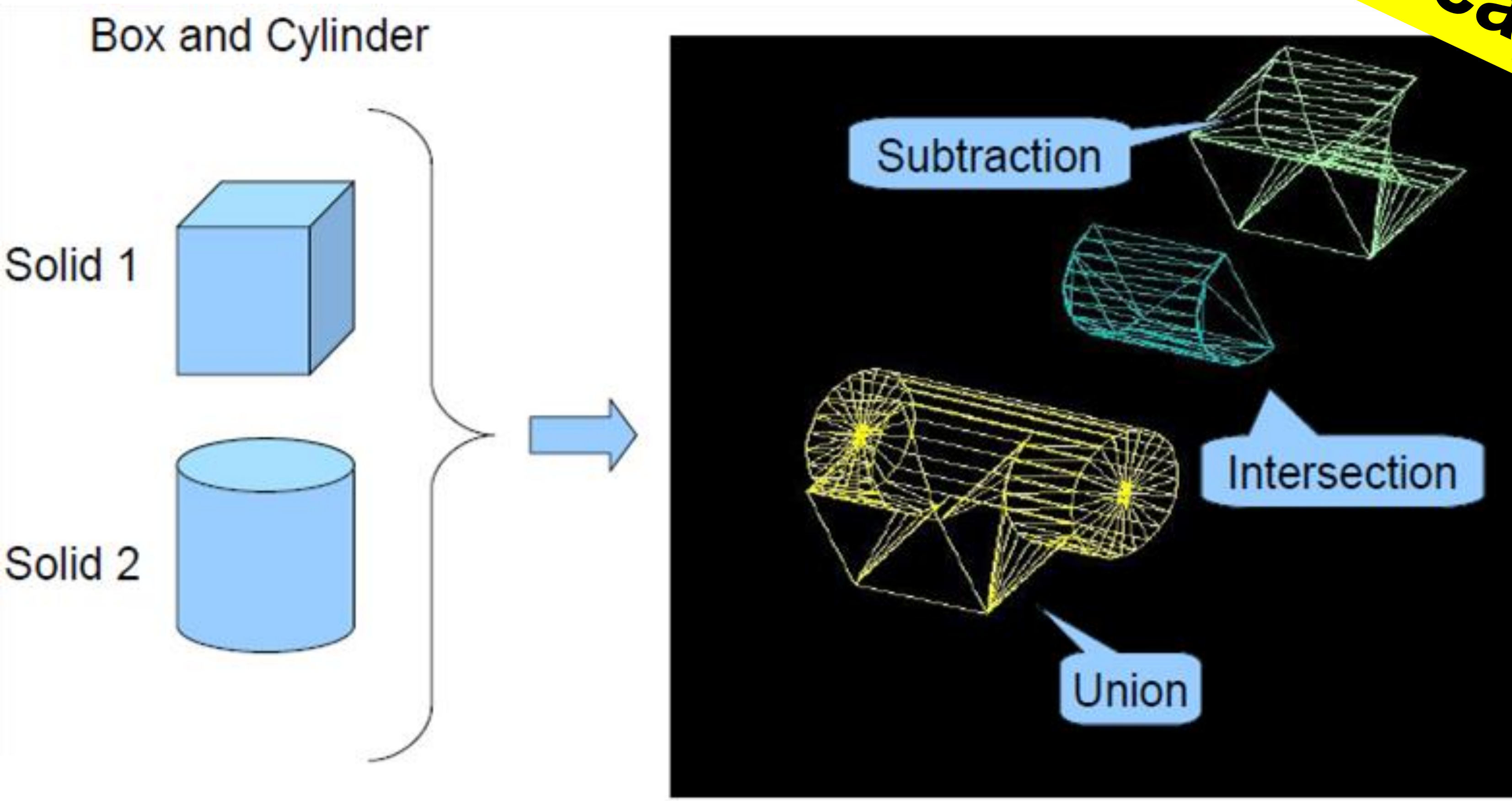
Topics

- overview, documentation (2020-10-29)
- geometry:
  - introduction, solids, logical/physical volumes (2020-11-05)
  - Boolean operations (2020-11-12)
  - multiple placements, hierarchy of volumes (2020-11-19)
  - replicas (2020-11-26)
- materials (2020-12-03)
- particles, processes (2020-12-17)
- user actions (2021-01-07)
- particle sources (2021-01-14)
- sensitive detectors (2021-01-21)

- evaluation test (2020-12-10, 2021-01-07)

Solids Made by Boolean Operations

Recap



Composing Solids Step by Step

Recap

1. Define only solid 1
   (mathematical shape of predefined CSG or previous Boolean operation)
   You do not need logical volume and placement!

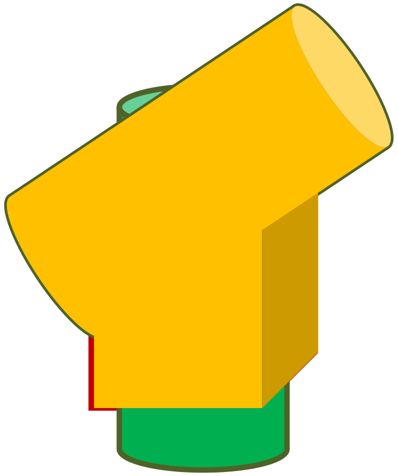
2. Define only solid 2
   (mathematical shape of predefined CSG or previous Boolean operation)
   You do not need logical volume and placement!

3. Translation and Rotation of solid 2

4. Boolean operation to make a new solid

5. Logical Volume for the new solid

6. Physical Volume (placement) for the new solid



Topics

- overview, documentation (2020-10-29)
- geometry:
  - introduction, solids, logical/physical volumes (2020-11-05)
  - Boolean operations (2020-11-12)
  - multiple placements, hierarchy of volumes (2020-11-19)
  - replicas (2020-11-26)
- materials (2020-12-03)
- particles, processes (2020-12-17)
- user actions (2021-01-07)
- particle sources (2021-01-14)
- sensitive detectors (2021-01-21)

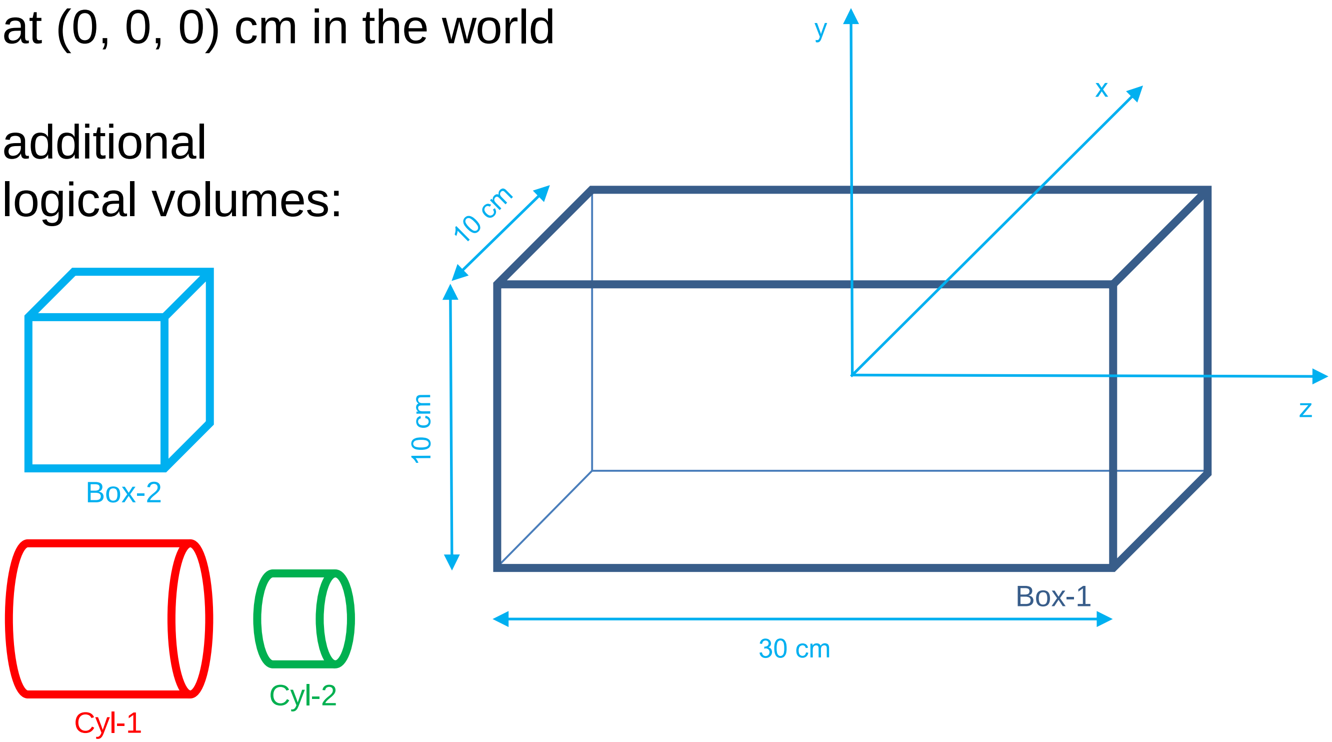
- evaluation test (2020-12-10, 2021-01-07)

Hierarchy of Volumes

Recap

logical volume Box-1 with 10 cm × 10 cm × 30 cm
at (0, 0, 0) cm in the world

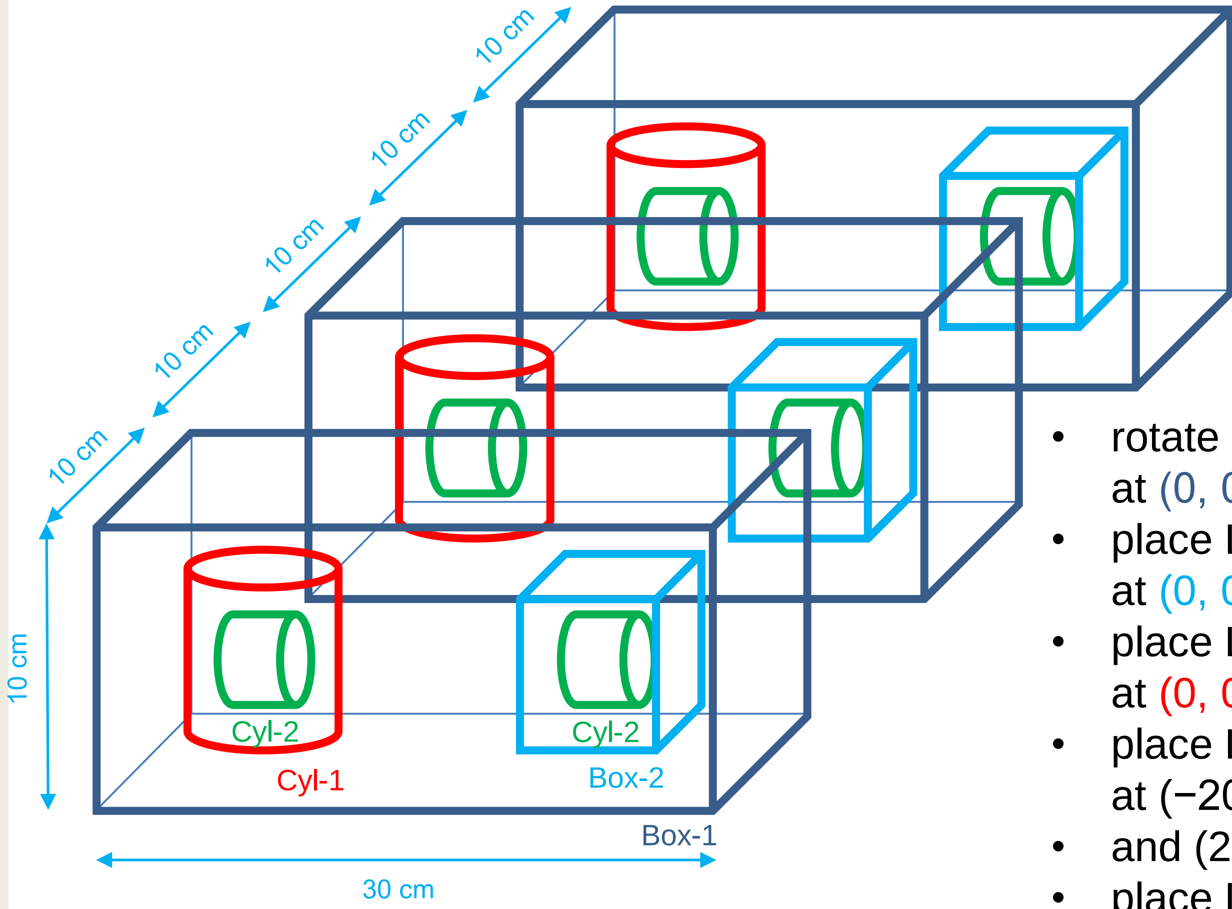
additional
logical volumes:



- Box-2: 5 cm × 5 cm × 5 cm
- Cyl-1: diameter 5 cm and height 5 cm
- Cyl-2: diameter 2 cm and height 2 cm

Multiple Placements of Volumes

Recap



- rotate and place LV Cyl-1 at (0, 0, −10) cm in LV Box-1
- place LV Cyl-2 at (0, 0, 0) cm in LV Box-2
- place LV Cyl-2 at (0, 0, 0) cm in LV Cyl-1
- place LV Box-1 at (−20, 0, 0) cm
- and (20, 0, 0) cm in the world
- place LV Box-2 at (0, 0, 10) cm in LV Box-1

Topics

- overview, documentation (2020-10-29)
- geometry:
  - introduction, solids, logical/physical volumes (2020-11-05)
  - Boolean operations (2020-11-12)
  - multiple placements, hierarchy of volumes (2020-11-19)
  - replicas (2020-11-26)
- materials (2020-12-03)
- particles, processes (2020-12-17)
- user actions (2021-01-07)
- particle sources (2021-01-14)
- sensitive detectors (2021-01-21)

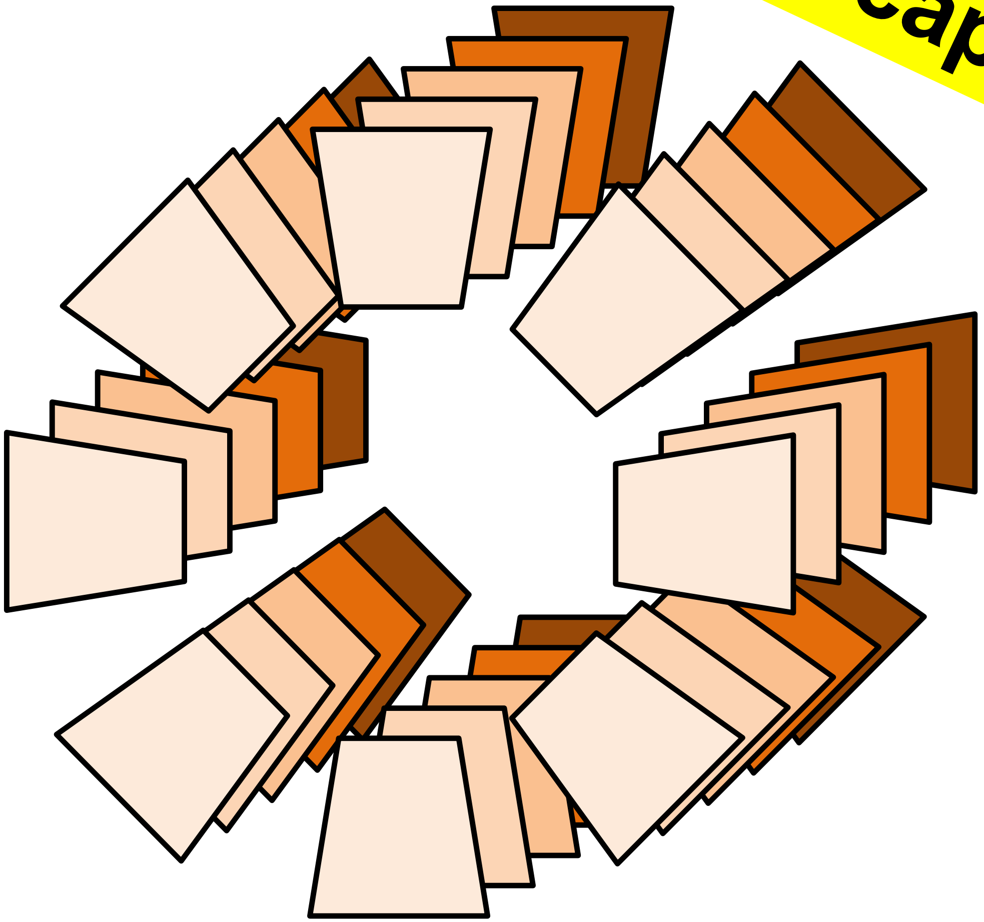
- evaluation test (2020-12-10, 2021-01-07)

Replicas

Recap

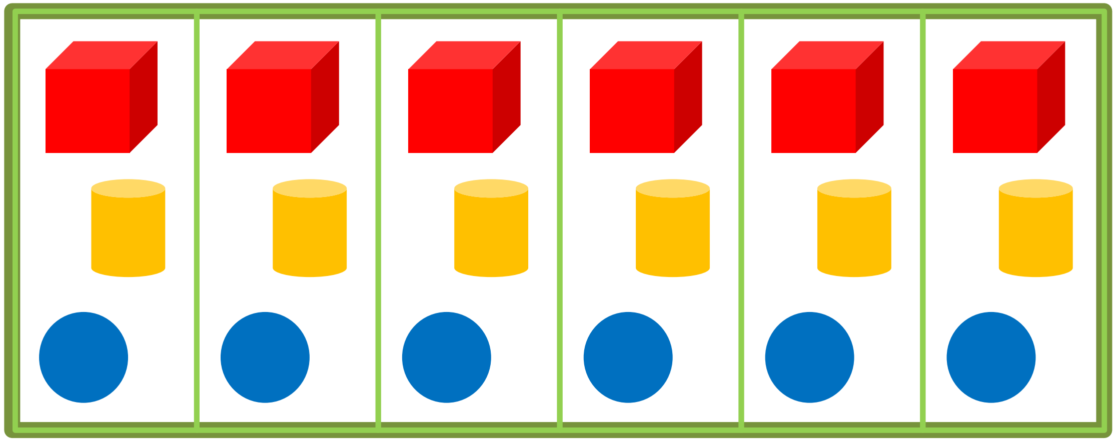
- easy way to repeat a given structure
- no need to manually calculate the individual coordinates of the objects

- G4PVReplica represents *n* volumes
  - differing only in their positioning
  - completely filling the containing mother volume
  - no gaps between repeated volumes
  - no other volumes in the mother volume



Replicas

Recap



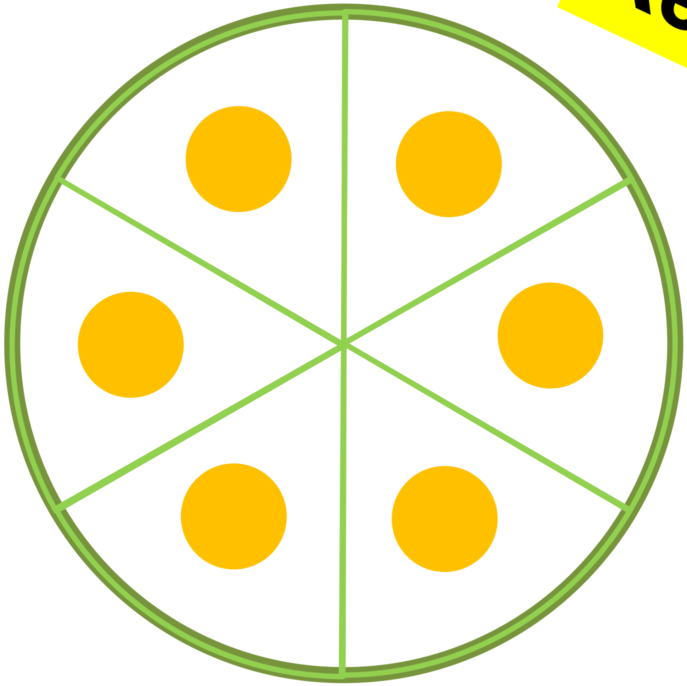
- mother volume (envelope) is divided into *n* equal daughter volumes (divisions)
- each division contains the same content

```
#include "G4PVReplica.hh"
repBoxes = new G4PVReplica("linear array",  // name
                           logicalBox,      // daugther
                           logicalEnvelope, // mother
                           kZAxis,          // axis
                           6,               // number of replicas
                           10.*cm);         // width along axis
```

Replicas

Recap

- mother volume (envelope) is divided into *n* equal daughter volumes (divisions)
- each division contains the same content



```
#include "G4PVReplica.hh"
repWedges = new G4PVReplica("circular array", // name
                            logicalWedge,      // daugther
                            logicalEnvelope,   // mother
                            kPhi,              // axis
                            6,                 // number of replicas
                            M_PI/3.*rad,       // width along axis
                            0);                // offset
```

Topics

- overview, documentation (2020-10-29)
- geometry:
  - introduction, solids, logical/physical volumes (2020-11-05)
  - Boolean operations (2020-11-12)
  - multiple placements, hierarchy of volumes (2020-11-19)
  - replicas (2020-11-26)
- materials (2020-12-03)
- particles, processes (2020-12-17)
- user actions (2021-01-07)
- particle sources (2021-01-14)
- sensitive detectors (2021-01-21)

- evaluation test (2020-12-10, 2021-01-07)

Materials, Elements, and Isotopes

Recap

The ***G4Element*** and ***G4Isotope*** classes describe the properties of the atoms:

- atomic number,
- number of nucleons,
- atomic mass,
- as well as quantities such as cross sections per atom, etc.

The ***G4Material*** class describes the macroscopic properties of matter:

- density,
- state,
- temperature,
- pressure,
- as well as macroscopic quantities like radiation length, mean free path, dE/dx, etc.

Elements with Isotopes

Recap

Define: Isotope 1 Isotope 2

^{235}U ^{238}U

Define an empty element with 2 components:

enriched uranium

…

…

Fill the element with isotopes (abundance in %):

enriched uranium

90% ^{235}U

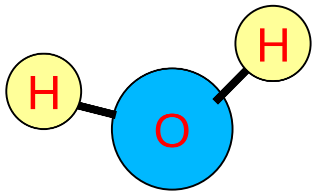
10% ^{238}U

Mixtures/Molecules

Recap

Example of Water Molecule

```
G4double Z, A, density;
G4int ncomps, natoms;
G4string symbol;
```




```
G4Element* el_H = new G4Element("Hydrogen", symbol="H", Z=1., A=1.01*g/mole);
G4Element* el_O = new G4Element("Oxygen",   symbol="O", Z=8., A=16.00*g/mole);
```

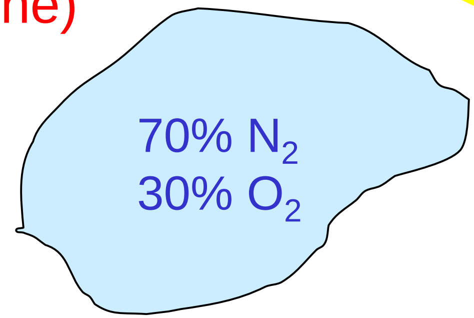
```
G4Material* mat_H2O = new G4Material("Water", density=1.000*g/cm3, ncomps=2);

mat_H2O->AddElement(el_H, natoms=2);
mat_H2O->AddElement(el_O, natoms=1);
```

Mixture by Fractional Mass

Recap

Example of Air (simple one)




```
G4double Z, A, density, fractionmass;
G4int ncomps;
G4string symbol;
```

```
G4Element* el_N = new G4Element("Nitrogen", symbol="N", Z= 7., A=14.01*g/mole);
G4Element* el_O = new G4Element("Oxygen",  symbol="O" , Z= 8., A=16.00*g/mole);
```

```
G4Material* mat_Air =  new G4Material("Air", density=1.290*mg/cm3, ncomps=2);

mat_Air->AddElement(el_N, fractionmass=0.7);
mat_Air->AddElement(el_O, fractionmass=0.3);
```

Mixture of Materials and Elements

Recap

Example of Aerogel (62.5 % SiO_2, 37.4% H_2O, 0.1% C)

```
G4double density, factionmass;
G4int ncomps;
```

```
G4Element* el_Si  = new G4Element(…);
G4Element* el_O  = new G4Element(…);
G4Element* el_H  = new G4Element(…);
G4Element* el_C  = new G4Element(…);
```

```
G4Material* mat_SiO2  = new G4Materia(…);
mat_SiO2->AddElement(…); …
G4Material* mat_H2O  = new G4Material(…);
mat_H2O->AddElement(…); …
```

```
G4Material* mat_Aerog = new G4Material("Aerogel", density=0.200*g/cm3, ncomps=3);
mat_Aerog->AddMaterial(mat_SiO2,  fractionmass=62.5*perCent);
mat_Aerog->AddMaterial(mat_H2O,   fractionmass=37.4*perCent);
mat_Aerog->AddElement(el_C,       fractionmass=  0.1*perCent);
```

Database of Materials and Elements

Recap

Using NIST (National Institute of Standards and Technology) Da

```
#include "G4Material.hh"
#include "G4NistManager.hh"
```

```
G4NistManager* man = G4NistManager::Instance();
```

```
// define pure NIST materials
G4Material* Al = man->FindOrBuildMaterial("G4_Al");
G4Material* Cu = man->FindOrBuildMaterial("G4_Cu");
```

```
// define NIST materials
G4Material* H2O = man->FindOrBuildMaterial("G4_WATER");
G4Material* SiO2 = man->FindOrBuildMaterial("G4_SILICON_DIOXIDE");
G4Material* Air = man->FindOrBuildMaterial("G4_AIR");
```

Topics

- overview, documentation (2020-10-29)
- geometry:
  - introduction, solids, logical/physical volumes (2020-11-05)
  - Boolean operations (2020-11-12)
  - multiple placements, hierarchy of volumes (2020-11-19)
  - replicas (2020-11-26)
- materials (2020-12-03)
- particles, processes (2020-12-17)
- user actions (2021-01-07)
- particle sources (2021-01-14)
- sensitive detectors (2021-01-21)

- evaluation test (2020-12-10, 2021-01-07)

Particles (Bosons & Leptons)

Recap

```
//Construct Bosons
// gamma
  G4Gamma::GammaDefinition();
// optical photon
  G4OpticalPhoton::OpticalPhotonDefinition();
```

```
  // leptons
  G4Electron::ElectronDefinition();
  G4Positron::PositronDefinition();
  G4MuonPlus::MuonPlusDefinition();
  G4MuonMinus::MuonMinusDefinition();

  G4NeutrinoE::NeutrinoEDefinition();
  G4AntiNeutrinoE::AntiNeutrinoEDefinition();
  G4NeutrinoMu::NeutrinoMuDefinition();
  G4AntiNeutrinoMu::AntiNeutrinoMuDefinition();
```

Particles

Recap

- easy way to instantiate all particles of a given category
  - G4BosonConstructor
  - G4LeptonConstructor
  - G4MesonConstructor
  - G4BaryonConstructor
  - G4ShortlivedConstructor
  - G4IonConstructor

```
// e.g. construct all leptons
 G4LeptonConstructor pConstructor;
 pConstructor.ConstructParticle();
```

Particles and Processes

Recap

In this tutorial we will use three particles:

Electrons

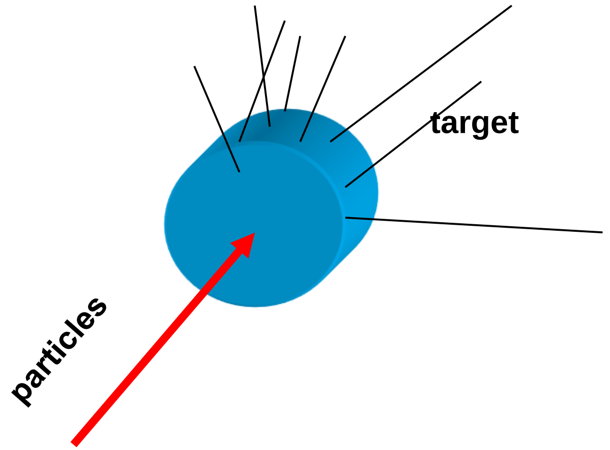
Positrons

Gamma

Which processes are needed ?

Color code of tracks in Geant4:

red	negatively charged particle
green	neutral particle
blue	positively charged particle



Processes

Recap

Physics processes describe how particles interact with materials.

Geant4 provides seven major categories of processes:

1. **transportation**
2. electromagnetic
3. decay
4. hadronic
5. optical
6. photolepton_hadron
7. parameterisation

Particles and Processes

Recap

If particle is a **Gamma** ⇒

- Photo Electric Effect → G4PhotoElectricEf
- Compton Scattering → G4ComptonScattering.hh
- Gamma Conversion → G4GammaConversion.hh

If particle is an **Electrons** ⇒

- Multiple Scattering → G4eMultipleScattering.hh
- Ionisation → G4eIonisation.hh
- Bremsstrahlung → G4eBremsstrahlung.hh

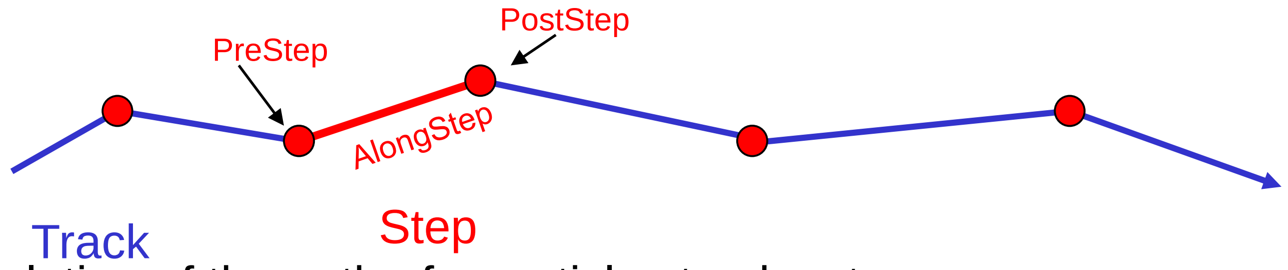
If particle is a **Positrons** ⇒

- Multiple Scattering → G4eMultipleScattering.hh
- Ionisation → G4eIonisation.hh
- Bremsstrahlung → G4eBremsstrahlung.hh
- Annihilation → G4eplusAnnihilation.hh

Processes

Recap

- each particle has its own G4ProcessManager providing a list of processes that this particle can undertake



- simulation of the path of a particle step by step

- three possibilities for processes to take place
  - at rest G4VProcess::AtRestDoIt
  - along step G4VProcess::AlongStepDoIt
  - post step G4VProcess::PostStepDoIt

- DoIt methods of the process class performs the physics processes:
  - momentum change
  - production of secondary particles

Processes

Recap

- registration of processes in G4ProcessManager is complex
- relations between processes are crucial in some cases
- easy way: G4PhysicsListHelper
  - users do not need to know about
    type of processes (at rest, discrete, continuous) and ordering

```
void MyPhysicsList::ConstructProcess() {
 AddTransportation();        // define transportation process
 ConstructEM();              // electromagnetic processes
}

void MyPhysicsList::ConstructEM() {
 G4PhysicsListHelper* ph = G4PhysicsListHelper::GetPhysicsListHelper(); // get pointer to helper

 G4ParticleDefinition* particle = G4Gamma::GammaDefinition();           // get pointer to gamma

 // construct and register processes for gamma
 ph->RegisterProcess(new G4PhotoElectricEffect(), particle);
 ph->RegisterProcess(new G4ComptonScattering(), particle);
 ph->RegisterProcess(new G4GammaConversion(), particle);
 ph->RegisterProcess(new G4RayleighScattering(), particle);
}
```

Topics

- overview, documentation (2020-10-29)
- geometry:
  - introduction, solids, logical/physical volumes (2020-11-05)
  - Boolean operations (2020-11-12)
  - multiple placements, hierarchy of volumes (2020-11-19)
  - replicas (2020-11-26)
- materials (2020-12-03)
- particles, processes (2020-12-17)
- user actions (2021-01-07)
- particle sources (2021-01-14)
- sensitive detectors (2021-01-21)

- evaluation test (2020-12-10, 2021-01-07)

Actions in Geant4

Recap

User classes

Initialization classes

Invoked at the initialization

G4VUserDetectorConstruction → Material and Geometry

G4VUserPhysicsList → Particles and Processes

Action classes

Invoked during an event loop

G4VUserPrimaryGeneratorAction → Primary Particles

G4UserRunAction

G4UserEventAction

G4UserSteppingAction

G4UserStackingAction

G4UserTrackingAction

Interaction with the simulation

main()

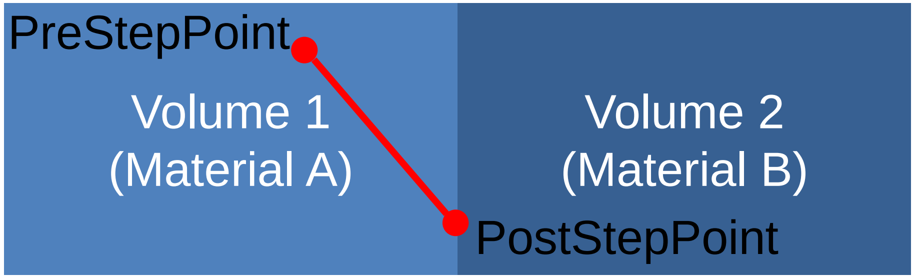
Geant4 does not provide main().

Note: classes written in Red are mandatory.

Steps in Geant4

Recap

- Step:
  - two points (PreStepPoint and PostStepPoint)
    - each point knows its volume and thus its material
    - if step is limited by a volume boundary:
      - end point is at the boundary and logically belongs to the next volume
    - simulation of boundary processes such as transition radiation or refraction
  - “delta” information of a particle
    - energy loss on the step
    - time of flight spent by the step
    - …



Topics

- overview, documentation (2020-10-29)
- geometry:
  - introduction, solids, logical/physical volumes (2020-11-05)
  - Boolean operations (2020-11-12)
  - multiple placements, hierarchy of volumes (2020-11-19)
  - replicas (2020-11-26)
- materials (2020-12-03)
- particles, processes (2020-12-17)
- user actions (2021-01-07)
- particle sources (2021-01-14)
- sensitive detectors (2021-01-21)

- evaluation test (2020-12-10, 2021-01-07)

Particle Gun

Recap

- Properties of G4ParticleGun can be modified event by event. Use the fol methods in GeneratePrimaries():

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

- GeneratePrimaryVertex() can be invoked more than once to generate additional particle tracks. Different particle properties are possible.
- More than one G4ParicleGun can be used in the primary generator action.

- Complex particle sources are possible.

Other Particle Generators

Recap

- G4GeneralParticleSource
  - it is used the same way as G4ParticleGun (globally replace G4ParticleGun with G4GeneralParticleSource)
  - configuration via methods and via command line/macro
  - specification of spectral, spatial, and angular distributions of the primary source particles
    - spectrum: mono-energetic, linear, exponential, power-law, Gaussian, bremsstrahlung, blackbody, cosmic diffuse gamma ray, or piece-wise fits to data
    - spatial sampling:
      - point source
      - planar sources: circles, annuli, ellipses, squares, or rectangles
      - 1D or 2D beam spots
      - surface or volume sources: sphere, ellipsoid, cylinder, or parallelepipedon
    - angular distribution: unidirectional, isotropic, cosine-law, beam, or arbitrary (user defined)
    - multiple sources: multiple independent sources can be used in the same run

- G4HEPEvtInterface
  - almost all HEP (High Energy Physics) event generators can store the generated events in ASCII files
  - Geant4 can read these ASCII files and can produce G4PrimaryParticle objects associated with a G4PrimaryVertex object.

Topics

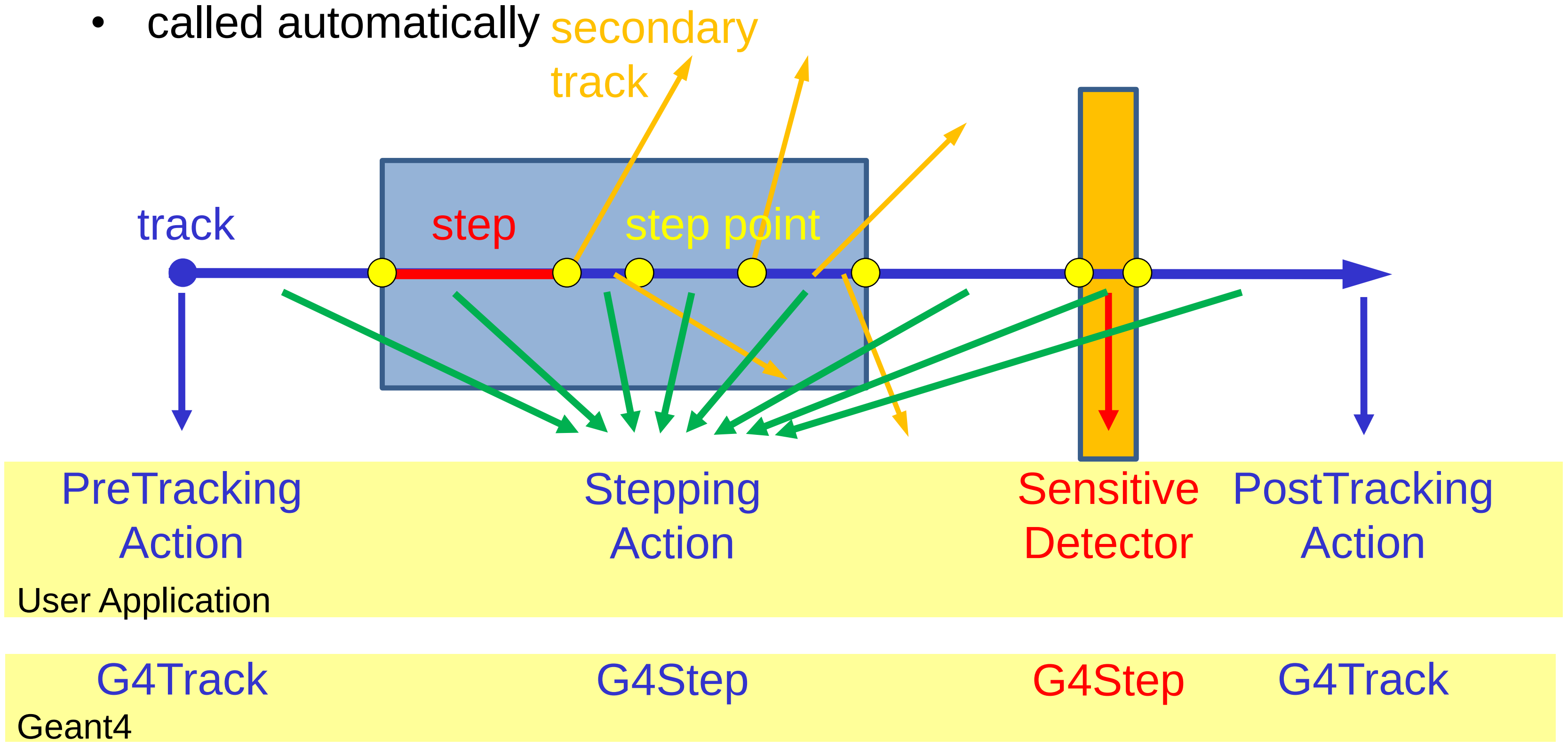
- overview, documentation (2020-10-29)
- geometry:
  - introduction, solids, logical/physical volumes (2020-11-05)
  - Boolean operations (2020-11-12)
  - multiple placements, hierarchy of volumes (2020-11-19)
  - replicas (2020-11-26)
- materials (2020-12-03)
- particles, processes (2020-12-17)
- user actions (2021-01-07)
- particle sources (2021-01-14)
- sensitive detectors (2021-01-21)

- evaluation test (2020-12-10, 2021-01-07)

Event Processing: Sensitive Detector

Recap

- special user class: sensitive detector
  - attached to logical volume
  - called automatically



Sensitive Detector Class

Recap

- sensitive detector class MySD is derived from G4VSensitiveDetector
  - three user methods called during event processing:
    - at beginning of each event: Initialize(…)
    - in a step (if in the associated volume): ProcessHits(…)
      - 1st argument: G4Step object
        - see stepping action
      - 2nd argument: G4TouchableHistory object
        - Readout geometry (or NULL)
      - process hits and fill them into hit collection
    - at end of each event: EndOfEvent(…)

Multi-functional Detector

Recap

- very generic detector class available: G4MultiFunctionalDetector
- use scorer classes derived from G4VPrimitiveScorer in order to collect specific data
  - track length G4PSTrackLength
  - passage track length G4PSPassageTrackLength
  - energy deposition G4PSEnergyDeposit
  - dose deposition G4PSDoseDeposit
  - flat surface current G4PSFlatSurfaceCurrent
  - sphere surface current G4PSSphereSurfaceCurrent
  - passage current G4PSPassageCurrent
  - flat surface flux G4PSFlatSurfaceFlux
  - cell flux G4PSCellFlux
  - passage cell flux G4PSPassageCellFlux
  - minimum kinetic energy of secondary particles G4PSMinKinEAtGeneration
  - number of secondary particles G4PSNofSecondary
  - number of steps G4PSNofStep
  - total charge of particles stopped G4PSCellCharge

Multi-functional Detector

Recap

- scorer classes above produce one G4THitsMap<G4double> object per e
  - mapping copy number of the volume to measured quantity
  - parameter depth determines which copy number is used (useful for replicated structures)
    - depth = 0: copy number of the physical volume itself
    - depth = 1: copy number of the mother volume
    - depth = 2: copy number of the mother volume of the mother volume
    - …
- collected data can be filtered using classes derived from G4VSDfilter
  - all charged particles G4SDChargedFilter
  - all neutral particles G4SDNeutralFilter
  - particles of given species G4SDParticleFilter
  - particles in a given kinetic energy range G4SDKineticEnergyFilter
  - given species in a given energy range G4SDParticleWithEnergyFilter

Topics

- overview, documentation (2020-10-29)
- geometry:
  - introduction, solids, logical/physical volumes (2020-11-05)
  - Boolean operations (2020-11-12)
  - multiple placements, hierarchy of volumes (2020-11-19)
  - replicas (2020-11-26)
- materials (2020-12-03)
- particles, processes (2020-12-17)
- user actions (2021-01-07)
- particle sources (2021-01-14)
- sensitive detectors (2021-01-21)

- evaluation test (2020-12-10, 2021-01-07)

Exam

Recap

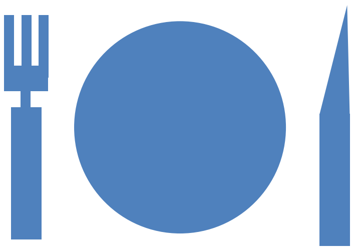
- What you should know and what can occur:
  - general principles and concepts in Geant4
  - capabilities of Geant4
  - explaining a Geant4 code fragment
  - explaining the purpose of a known Geant4 class
  - understanding of the underlying physical principles
  - completing a short simple code fragment
  - choosing the correct code fragment from given alternatives
  - explaining how to build a given geometry, material, … in terms of necessary steps (no line of code necessary)

- What will not occur:
  - a complex task like this evaluation test
  - writing a functioning piece of code without any help
  - exact knowledge of the spelling of names of Geant4 classes and methods

Exam: Examples

Recap

- How do you build the material CsI(Tl 5%) in Geant4?
- How do you build this geometry?
  (all parts have the same thickness)



- What is the purpose of this code fragment?

```
void DetectorPhysEventAction::BeginOfEventAction(const G4Event* aEvt) {
  G4cout << "Start of Event" << G4endl;
}
```

Final Remarks

- Geant4
  - in this tutorial: only basic concepts and examples
  - very powerful toolkit
  - many fields of application

- other important advanced topics
  - particle propagation in electromagnetic fields
  - simulation of digitization of hits in detector components
  - interface to Root histograms and trees
  - customization of user interface, own macro commands to steer the simulation
  - using macro files to run a series of simulations with different settings (particle energy, geometrical set-up, …)
  - interface to HEP event generators
  - different visualization options
  - multithreading, make use of several CPU cores

- Have a look into the rich collection of Geant4 examples!