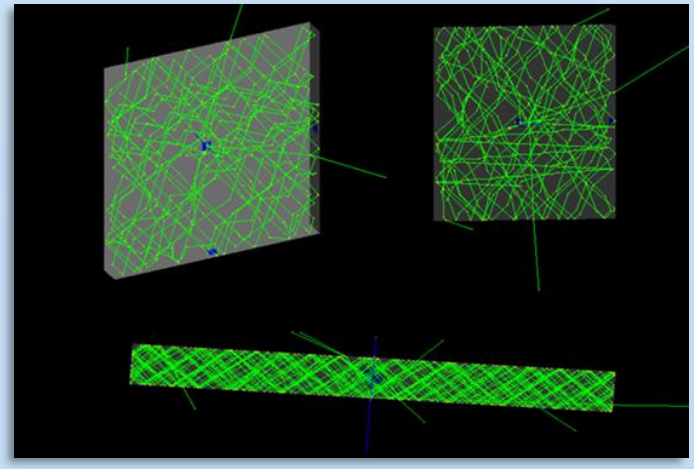
Simulation of optical photon propagation for generic scintillator-based detectors

Lecture 3

Geometry



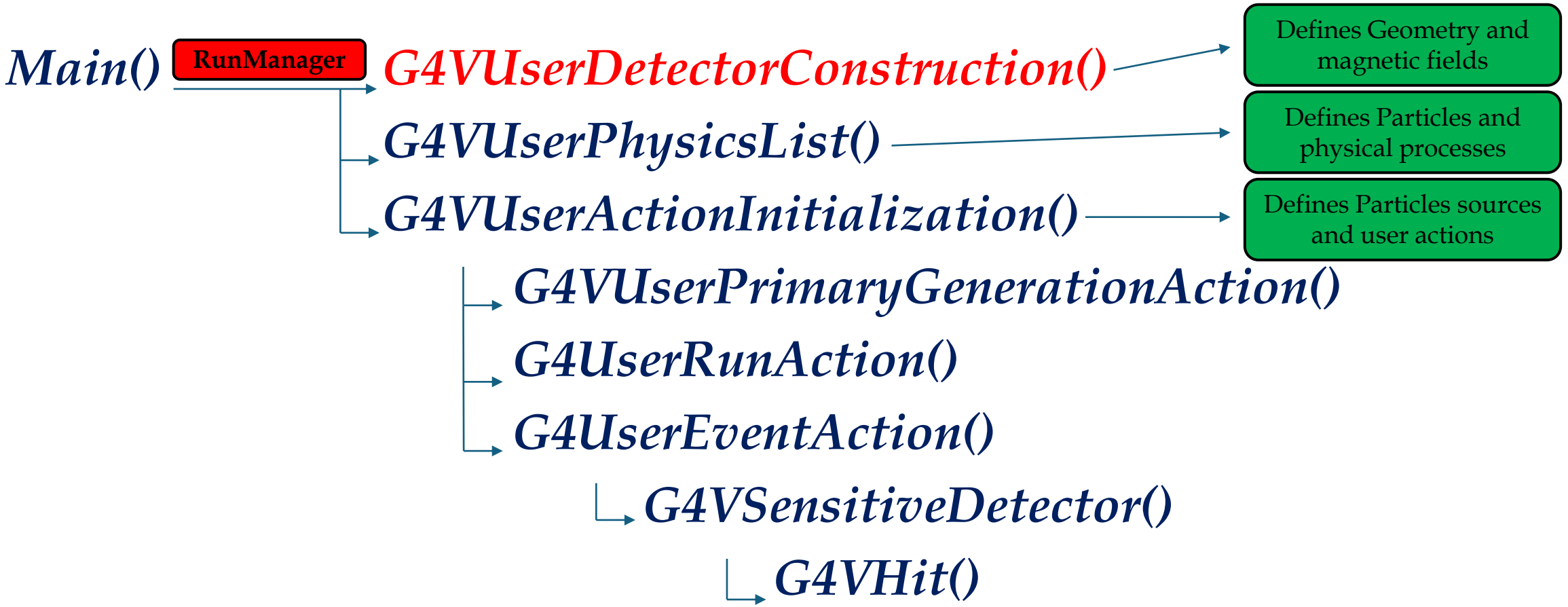
Davide Serini
davide.serini@ba.infn.it

Hands-on schedule

- General G4 introduction
  - How to run the example B1
  - Build and run the code
    - *Run a simple simulation using the User Interface*
    - *Run a simple simulation using a macro*

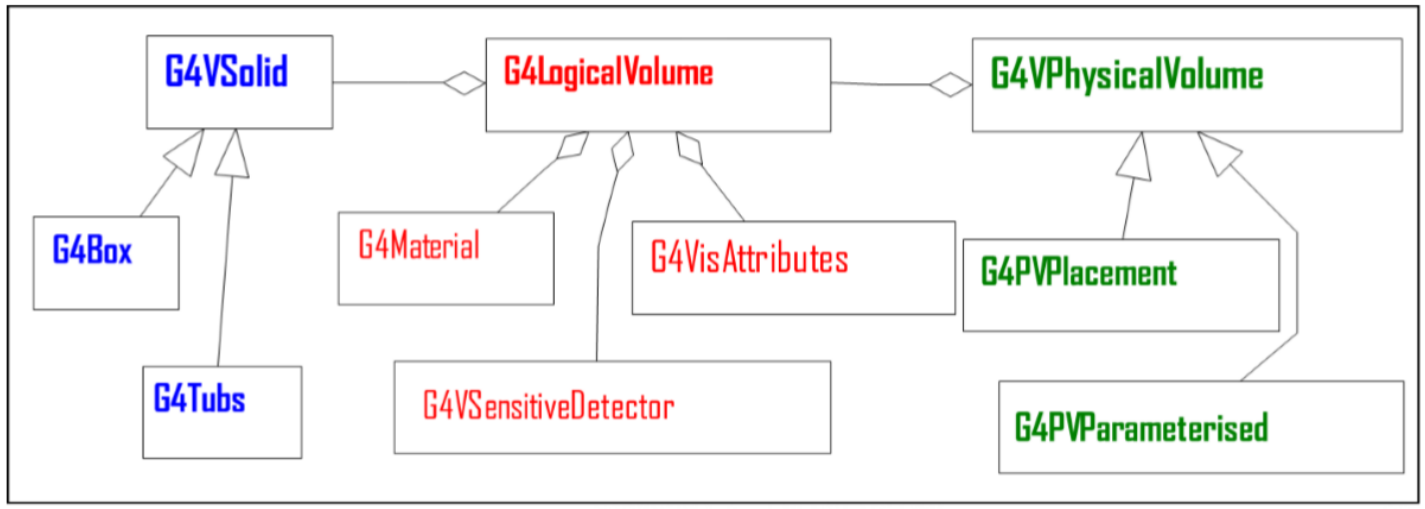
- Make your own project
  - Build your basic project:
    - *main & physics list*
    - *geometry*
    - *write a macro*
  - Build your simulation: data management tools:
    - *Sensitive Detector*
    - *Hit collection*
    - *Run Action*
    - *Event Action*
  - Make your simulation (final exercise)

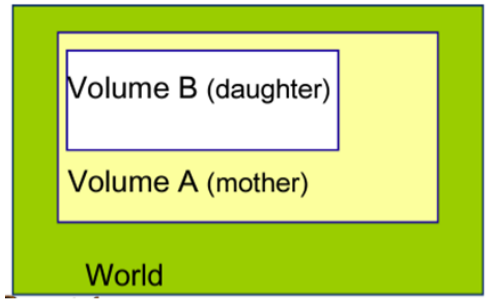
Classes hierarchy



Detector Construction overview

- DetectorConstruction() class inherited from G4VUserDetectorConstruction()
  - G4VSolid()
    - *Defines shape and size of basic geometric solids*
  - G4VLogicalVolume()
    - *Defines material, sensitivity, magnetic field, visual attributes*
  - G4VPhysicalVolume()
    - *Defines hierarchy of volumes, position, rotations, different copies of the same logical volume…*
- A unique physical volume (the world volume), which represents the experimental area, must exist and fully contain all other components





e.g.: Volume A is mother of Volume B

The mother must contain the daughter volume entirely

Detector Construction overview

- Defines the shape of a volume
  - *Multiple ways: from geometric primitives to complex meshes*
- Defines the size of a volume
  - *Don't forget to give units!*

`Solid`

Start with describing its shape:
- Box 5m x 2m x 10m
- Sphere r=10m
- Cylinder r = 5m, h = 10 m …

- Attach a material to a solid
- Make a solid sensitive
- Define visualization attributes for a solid
- ***Add physical daughter volumes***

`Logical Volume`

Which properties does it have?
- Material
- Is it sensitive, i.e. used for read-out?

- Set the location and the rotation of a volume
- Generate replicas for repeated volumes

`Physical Volume`

Finally, place it:
Single placement or replicas?

System of Units

- Geant4 has no default unit and does not deduce the unit for you
  - If you define a number you should also provide a unit by multiplying it to the number.
  - If no unit is given Geant4 uses an internal unit. This is strongly discouraged. Almost all commonly used units are available
  - To retrieve a value with a specific unit you divide by this unit
  - You can also let Geant4 choose the most appropriate unit to output in by giving it a "unit" category

```
G4double length = 1.0*m;
G4double density = 100*g/cm3
G4double magnetic_field = 2.*Tesla;
```

```
G4double length = 1.0*m;
G4cout<<"Length: "<<length/km<<" km"<<G4endl;
G4cout<<"Length: "<<length/cm<<" cm"<<G4endl;
G4cout<<"Length: "<<length/nm<<" nm"<<G4endl;
```

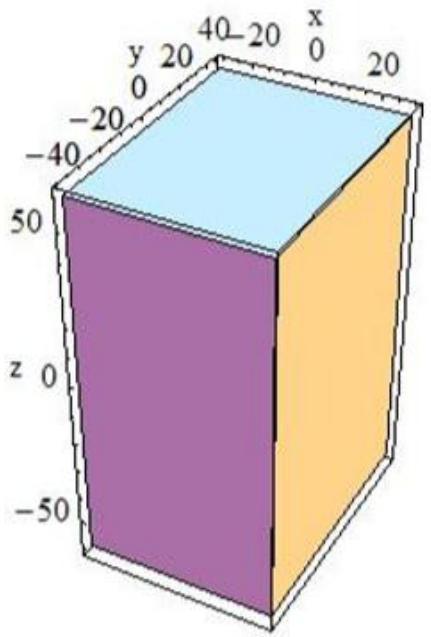
```
G4double dE= 100*keV;
G4cout<<G4BestUnit(dE, "Energy")<<G4endl;
//will print dE in MeV, keV, eV depending on value
```

Example of G4VSolid: G4Box

- A complete guides for Solid Geometry is available in Application Developers Guide, Chapter 4.1.2

```
G4Box(const G4String& pName,
            G4double  pX,
            G4double  pY,
            G4double  pZ)
```

pX,pY,pZ are half-lengths!!



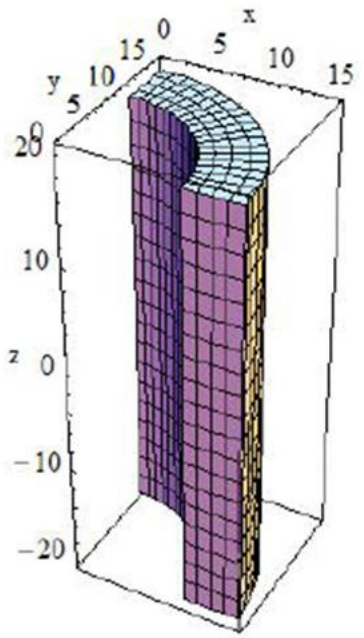
In the picture:
pX = 30, pY = 40, pZ = 60

pX	half length in X	pY	half length in Y	pZ	half length in Z

Example of G4VSolid: G4Tubs

- A complete guides for Solid Geometry is available in Application Developers Guide, Chapter 4.1.2

```
G4Tubs(const G4String& pName,
             G4double  pRMin,
             G4double  pRMax,
             G4double  pDz,
             G4double  pSPhi,
             G4double  pDPhi)
```

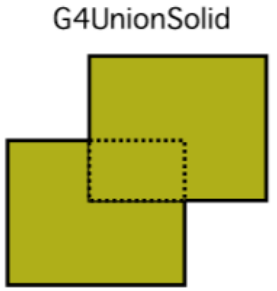


In the picture:
`pRMin = 10, pRMax = 15, pDz = 20`

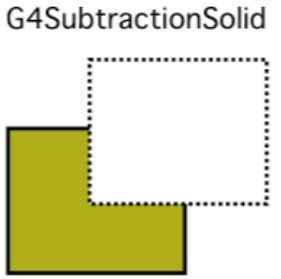
`pRMin`	Inner radius	`pRMax`	Outer radius
`pDz`	Half length in Z	`pSPhi`	Starting phi angle in radians
`pDPhi`	Angle of the segment in radians		

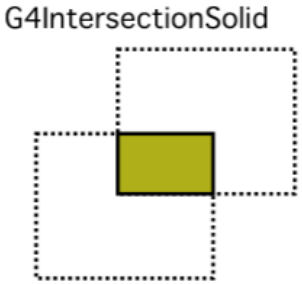
Example of G4VSolid: Boolean Solid

- Boolean solid:
  - G4Union Solid()
  - G4SubtractionSolid()
  - G4IntersectionSolid()



- Boolean operations allow combinations of two solids
  - 2nd solid is positioned (and optionally transformed) w.r.t. coordinate system of first solid
  - A new solid (which again can again participate in boolean operations)
  - Boolean operations allow for complex geometries with usually quite little effort.

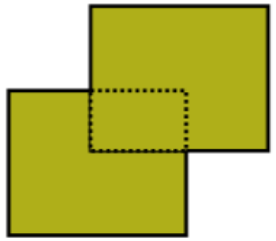




Example of G4VSolid: Boolean Solid

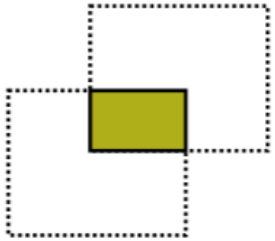
```
G4VSolid* boxSolid1 = new G4Box("aBox1", 1.0*m, 1.0*m, 1.0*m);
G4VSolid* boxSolid2 = new G4Box("aBox1", 0.5*m, 0.5*m, 0.5*m);


G4VSolid* BoxBoxSolid = new G4UnionSolid("aBoxBox",
        boxSolid1, boxSolid2, //two solids to operate on
        0, G4ThreeVector(0., 0., 95.*cm)); // translation

G4VSolid* hollowBoxSolid = new G4SubtractionSolid("aHollowBox",
        boxSolid1, boxSolid2, //two solids to operate on
        0, G4ThreeVector(0., 0., 5.*cm)); // translation

G4VSolid* BoxBoxSolid = new G4IntersectionSolid("aIntersectedBox",
        boxSolid1, boxSolid2, //two solids to operate on
        0, G4ThreeVector(0., 0.95*cm, 95.*cm)); // translation
```

G4UnionSolid



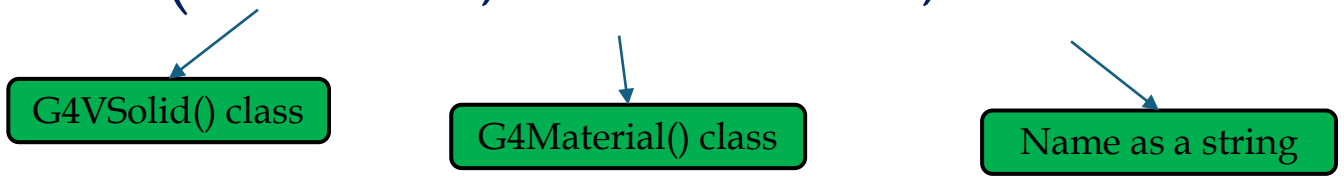
G4SubtractionSolid

G4IntersectionSolid



G4Logical Volume

G4LogicalVolume(SolidBox, BoxMaterial, "VolumeName")



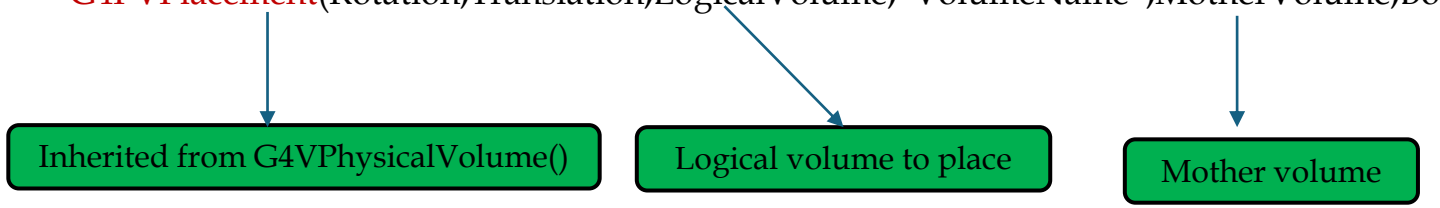
- Material can be created as mixture of elements or molecules knowing:
  - Density,pressure,…
- Native G4 materials are imported from NIST database (https://physics.nist.gov/PhysRefData/)
- G4NistManager()
  - FindOrBuildElement()
  - FindOrBuildMaterial()
  - ConstructNewMaterial()

Material example

```
G4NistManager *fNistManager = G4NistManager::Instance();
fNistManager->SetVerbose(0);

G4Material* world_mat    = fNistManager->FindOrBuildMaterial("G4_AIR"); // try G4_Galactic
G4Material* scint_mat    = fNistManager->FindOrBuildMaterial("G4_PLASTIC_SC_VINYLTOLUENE"); //
G4Material* SiPM_mat     = fNistManager->FindOrBuildMaterial("G4_Si");

//------------------------------------------------------------
// Here we create a new material "Teflon" for the Wrapping
//------------------------------------------------------------
G4double density;
std::vector<G4int>    natoms;
std::vector<G4String> elements;
G4int ncomponents;
G4double fractionmass;
elements.push_back("C");     natoms.push_back(2);
elements.push_back("F");     natoms.push_back(4);
density = 2.2*g/cm3;
G4Material* Teflon = fNistManager->ConstructNewMaterial("Teflon", elements, natoms, density);
elements.clear();
natoms.clear();
G4Material* wrapping_mat = Teflon;
//------------------------------------------------------
```

G4VPhysical Volume

G4PVPlacement(Rotation,Translation,LogicalVolume,"VolumeName",MotherVolume,BooleanOperation,CopyNumber,CheckOverlaps)



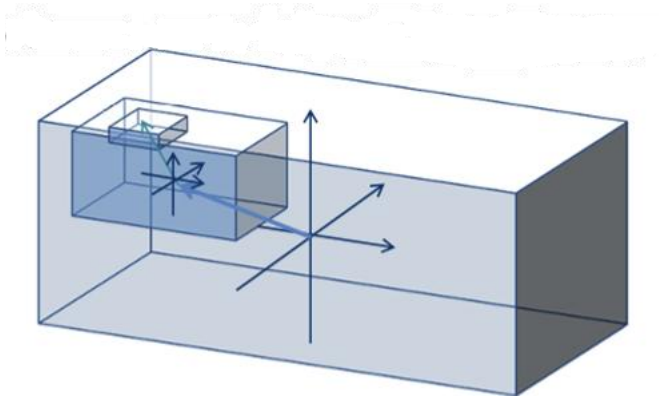
- Concept of Mother Volume:
  - The volume is placed inside another logical volume (**usually the World volume**), all the position and rotation are referred to the mother volume…
  - If the mother volume is copied, rotated or translated all the daughter volumes are also copied, rotated and translated…

  - **G4PVPlacement**(

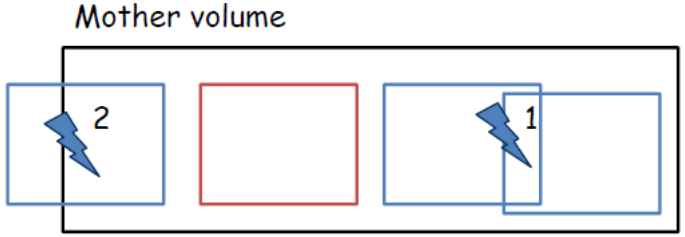
```
G4RotationMatrix* pRot,        // rotation w.r.t. to mother volume
constG4ThreeVector& trans,     // translation w.r.t. mother
G4LogicalVolume* pLog,         // solid logical volume constG4String& name,
G4LogicalVolume* pMLog,        // mother logical volume G4bool pMany, // not used
G4int copyNo,                  // set to 0 for first volume of type
G4bool checkOverlaps);         // check for overlaps
```

Geometrical Hierarchy

- Coordinate systems in Geant4 are defined by the respective mother volume
  - i.e. all daughter volumes are placed relative to their mother volume's local coordinate system
  - The origin of the mother volume's local coordinate system is the center of the volume A mother volume has to fully enclose its daughter volumes



In Geant4 volumes should not overlap on the same hierarchical level (1) and daughters should be fully enclosed in their mother volumes (2) Even if great care is taken, it is not improbable that one or both of these problems may occur when implementing a new geometry Geant4 and external tools help in finding such "mistake"



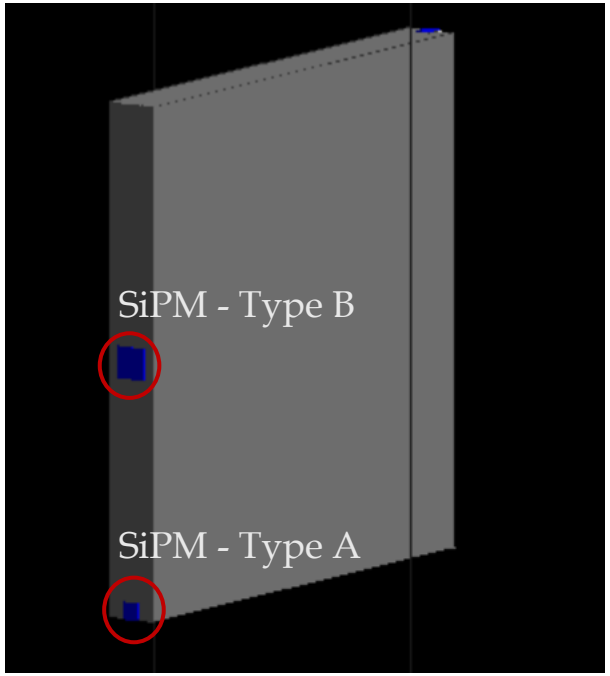
Example: Build your plastic scintillator tile

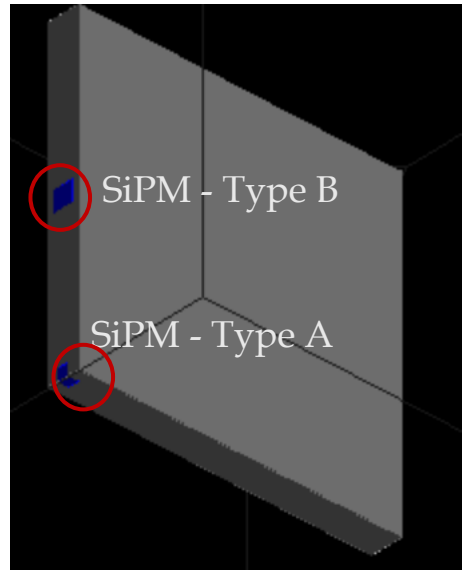
```
  //---------------//
  // *** World *** //
  //---------------//

  // - Creation and placement - //
  G4Box* solidWorld = new G4Box("World",0.5*world_sizeX, 0.5*world_sizeY, 0.5*world_sizeZ);
  G4LogicalVolume* logicWorld = new G4LogicalVolume(solidWorld,world_mat,"World");
  G4VPhysicalVolume* physicalWorld = new G4PVPlacement(0,G4ThreeVector(),logicWorld,"World",0,false,0,checkOverlaps);

  // - Visualization
  G4VisAttributes* WorldVis = new G4VisAttributes(G4Colour(0.1,0.1,0.1));
  //logicWorld->SetVisAttributes (G4VisAttributes::GetInvisible());
  WorldVis->SetVisibility(true);
  logicWorld->SetVisAttributes(WorldVis);
  //-------------------------------------------------------//

  //-----------------------//
  // *** Scintillators *** //
  //-----------------------//

  // - Creation and placement - //
  G4Box* solidScintillator = new G4Box("Scintillator",0.5*ScintillatorX, 0.5*ScintillatorY, 0.5*ScintillatorZ);
  G4LogicalVolume* logicScintillator = new G4LogicalVolume(solidScintillator,scint_mat,"Scintillator");
  G4VPhysicalVolume* physicalScintillator = new G4PVPlacement(0,G4ThreeVector(),logicScintillator,"Scintillator",logicWorld,true,0,checkOv
erlaps);

  // - Visualization
  G4VisAttributes* ScintillatorVis = new G4VisAttributes(G4Colour(0.9,0.9,0.9));
  ScintillatorVis->SetVisibility(true);
  ScintillatorVis->SetForceSolid(true);
  logicScintillator->SetVisAttributes(ScintillatorVis);
  //-------------------------------------------------------//
```

Exercise – Example0

- Starting from the Blank project create your plastic scintillator tile of $10x10x1cm^3$ equipped with:
  - 4 SiPMs (type A) of 3mm size on two edges of your tile
  - 2 SiPMs (type B) of 6mm size on the center of two small faces
- Wrap your tile with a 300 μm of Teflon
  - Suggestion: you need to «create the window» for your SiPMs (using multiple times G4Subtraction)

```
// World
G4double world_sizeX  = 20*cm;
G4double world_sizeY  = 20*cm;
G4double world_sizeZ  = 20*cm;

G4double ScintillatorX      = 10*cm;
G4double ScintillatorY      = 10*cm;
G4double ScintillatorZ      = 1*cm;
G4double SiPMThickness      = 0.4*mm;
G4double WrappingThickness  = 300*um;
G4double SiPMSize           = 3*mm;
G4double SiPMSize2          = 6*mm;
```





Links

Blank

wget 'https://istnazfisnucl-my.sharepoint.com/:u:/g/personal/serini_infn_it/EX-1XCKrJdlImqv1wKKZY20BiiMb9b5eAdP1ozxKD2wLyQ?e=Y7DJeZ&download=1' -O Blank.tar.gz

Ex0

wget 'https://istnazfisnucl-my.sharepoint.com/:u:/g/personal/serini_infn_it/EfRyYDwP7gZJruci7aTx1V4Bhpm-EaYmroRFsEnDIGLRTQ?e=PICleo&download=1' -O Ex0.tar.gz