(Hands-on)
Geometry Implementation

A.Kimura, Ashikaga Univ.
T.Aso, NIT-Toyama

Outline

- Learn
  - How to define the materials
  - How to implement the geometries

- You will do:
  - Ex1 Define materials from
    - The NIST predefined material-list
    - The user’s recipes
  - Ex2 Describe geometries using
    - CSG, LogicalVolume, and PhysicalVolume
    - Copy number
    - Daughter geometry
  - Ex3 Demonstrate a water-phantom using a sample voxelated geometry

Code Galet::DetectorConstruction

- Materials and geometries are defined in the **DetectorConstruction.cc**
  - The method **Construct()** in the class is called by the RunManager.
    - In the hands-on code, **the Construct()** method calls following two methods.

DetectorConstruction.cc

```
G4VPhysicalVolume* DetectorConstruction::Construct(){
  // Define materials
  DefineMaterials();
  // Define volumes
  return DefineVolumes();
}
```

  - In this hands-on, you will do:
    - Create materials in the **DefineMaterials()** method,
    - Create geometries in the **DefineVolumes()** method.

Exercise 1: Constructing materials

- Ex1-1 Create materials from NIST predefined database
  - Create G4_Al and G4_WATER
- Ex1-2 Create composite materials
  - Create Acrylic w/ element compositions $(C_5H_8O_2)$
- Ex1-3 Check the material parameters of created materials

Ex1-1 Create materials from NIST predefined database (1)

- Edit the code of DetectorConstruction.cc

  $ code ~/Galet-v11-MedEx-01/src/DetecrtorConstruction.cc

```
//....ooo00000ooo.......ooo00000ooo.......ooo00000ooo.......ooo00000ooo.....
G4VPhysicalVolume* DetectorConstruction::Construct(){
  // Define materials
  DefineMaterials();
  // Define volumes
  return DefineVolumes();
}
//....ooo00000ooo.......ooo00000ooo.......ooo00000ooo.......ooo00000ooo.....
void DetectorConstruction::DefineMaterials(){
  G4NistManager* nist = G4NistManager::Instance();
  nist->FindOrBuildMaterial("G4_AIR");

  //
  // Print materials
  G4cout << *(G4Material::GetMaterialTable()) << G4endl;
}
```

Ex1-1 Create materials from NIST predefined database (2)

- Insert lines to create G4_Al and G4_WATER

```
//....ooo00000ooo.........ooo00000ooo.........ooo00000ooo.........ooo00000ooo......
void DetectorConstruction::DefineMaterials(){
  G4NistManager* nist = G4NistManager::Instance();
  nist->FindOrBuildMaterial("G4_AIR");
  nist->FindOrBuildMaterial("G4_Al");
  nist->FindOrBuildMaterial("G4_WATER");

  //
  // Print materials
  G4cout << *(G4Material::GetMaterialTable()) << G4endl;
}
```

The NIST material database here

Ex1-2 Create materials from NIST predefined database

- Create the Acrylic using the element composition

Recipe
name: Acrylic
density:
1.19 g/cm^3
Elements:
$C_5H_8O_2$
Excitation energy:
74.0 eV

```
//....oooOOOOOooo.......oooOOOOOooo.......oooOOOOOooo.......oooOOOOOooo.....
void DetectorConstruction::DefineMaterials(){
  G4NistManager* nist = G4NistManager::Instance();
  nist->FindOrBuildMaterial("G4_AIR");
  nist->FindOrBuildMaterial("G4_Al");
  nist->FindOrBuildMaterial("G4_WATER");
  //
  G4Material* Acrylic = new G4Material("Acrylic", 1.19*g/cm3, 3 );
  Acrylic->AddElementByNumberOfAtoms(nist->FindOrBuildElement("C"),5);
  Acrylic->AddElementByNumberOfAtoms(nist->FindOrBuildElement("H"),8);
  Acrylic->AddElementByNumberOfAtoms(nist->FindOrBuildElement("O"),2);
  G4IonisParamMat* acrylicParam = Acrylic->GetIonisation();
  acrylicParam->SetMeanExcitationEnergy(74.0*eV);
  //
  // Print materials
  G4cout << *(G4Material::GetMaterialTable()) << G4endl;
}
```

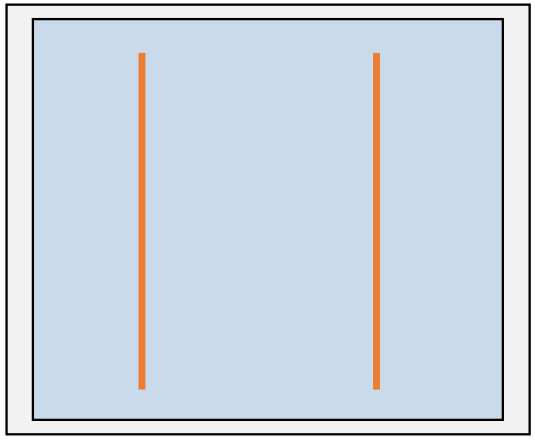
Ex1-3 Rebuild the application and check the material properties

- Save the DetectorConstruction.cc （Keep the vscode window, we will use it again,）
- Click the terminal window to be active and build the Galet
  $ cd ~/Galet-v11-MedEx-01-build
  $ make
  - If there are errors, maybe there is a spelling mistake in the code.
  - Check your code again according to the error messages.
  - Save the code again, and then make it again.
- Check the created materials in the Galet w/o GUI
  $ ./Galet -u tcsh
  Idle> /material/g4/printMaterial G4_Al
  Idle> /material/g4/printMaterial G4_WATER
  Idle> /material/g4/printMaterial Acrylic

Acrylic

```
Idle> /material/g4/printMaterial Acrylic
/material/g4/printMaterial Acrylic
 Material:  Acrylic    density:  1.190 g/cm3   RadL:  34.535 cm   Nucl.Int.Length:  63.846 cm
                       Imean:  74.000 eV   temperature: 293.15 K  pressure:   1.00 atm

   --->  Element: C (C)   Z =  6.0   N =    12   A = 12.011 g/mole
         --->  Isotope:   C12   Z =  6   N =  12   A =  12.00 g/mole   abundance: 98.930 %
         --->  Isotope:   C13   Z =  6   N =  13   A =  13.00 g/mole   abundance:  1.070 %
          ElmMassFraction:  70.57 %  ElmAbundance  44.44 %

   --->  Element: H (H)   Z =  1.0   N =     1   A =  1.008 g/mole
         --->  Isotope:    H1   Z =  1   N =   1   A =   1.01 g/mole   abundance: 99.989 %
         --->  Isotope:    H2   Z =  1   N =   2   A =   2.01 g/mole   abundance:  0.011 %
          ElmMassFraction:   5.92 %  ElmAbundance  44.44 %

   --->  Element: O (O)   Z =  8.0   N =    16   A = 15.999 g/mole
         --->  Isotope:   O16   Z =  8   N =  16   A =  15.99 g/mole   abundance: 99.757 %
         --->  Isotope:   O17   Z =  8   N =  17   A =  17.00 g/mole   abundance:  0.038 %
         --->  Isotope:   O18   Z =  8   N =  18   A =  18.00 g/mole   abundance:  0.205 %
          ElmMassFraction:  23.50 %  ElmAbundance  11.11 %

Idle>
```

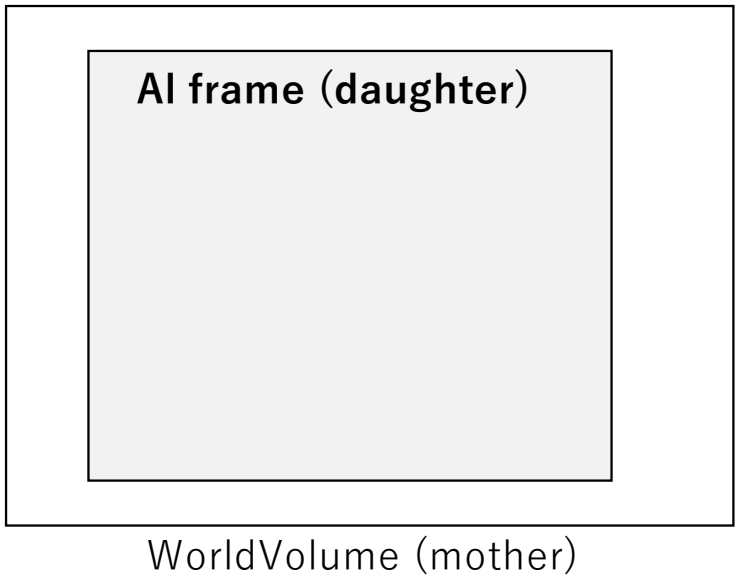
Ex2 Create a cylindrical monitor (Dummy monitor)



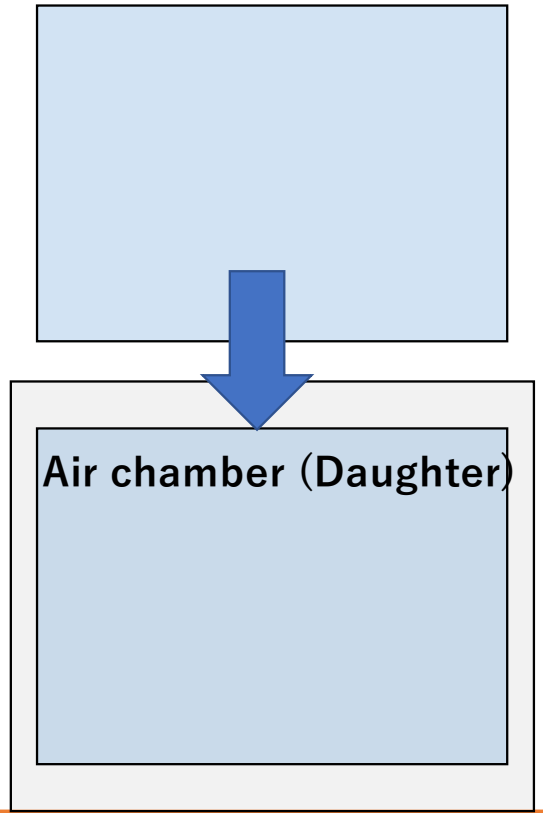
- Parameters
  - The outer envelope
    - Name: Frame
    - Rin 0 mm, Rout = 145 mm, dzHalf = 30 mm
    - Material G4_Al
    - Placement (0, 0, 1450.) mm in the Word Volume system
  - Inner chamber
    - Name: Chamber
    - Rin = 0 mm, Rout = 140 mm, dzHalf = 29.96 mm
    - Material G4_AIR
    - Placement (0, 0, 0) mm in the Frame (The Frame is a daughter volume of the Frame)
  - Electrodes
    - Name: Elec
    - Rin - 0 mm, Rout - 125 mm, dzHalf = 0.0075 mm
    - Material G4_Al
    - Placement (0, 0, -10.), (0, 0, +10) mm in the Chamber system by using the same logical volume with the copy numbers of 0 and 1. (Copy geometry) (The Elecs are daughter volumes of the Chamber.)

Ex2 Create a cylindrical monitor (Dummy monitor)

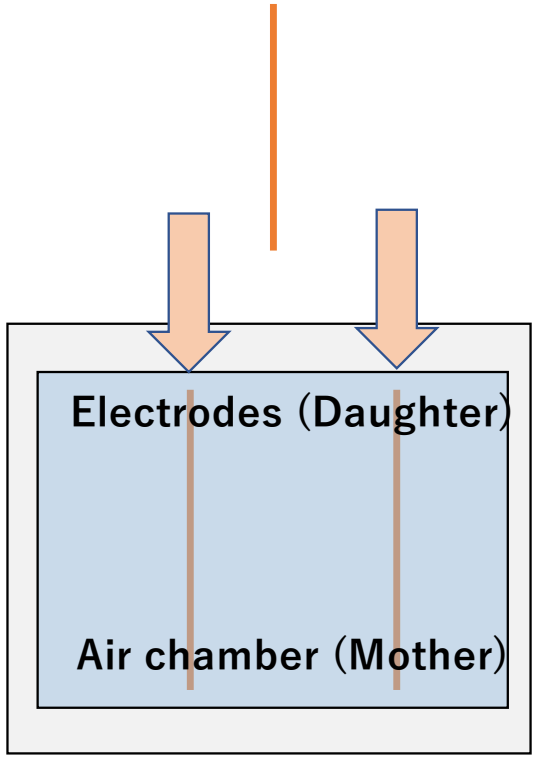
(1) Create an Al frame as a cylinder and place it in the WorldVolume



(2) Create an Air chamber as a cylinder and place it in the Al frame



(3) Create an Al electrode as a thin cylinder and place it in the Air chamber twice



Ex2 Create a cylindrical monitor (Dummy monitor) (0)

- Edit the code of DetectorConstruction.cc

If you have closed the vscode window,
$ code ~/Galet-v11-MedEx-01/src/DetecrtorConstruction.cc

```
G4VPhysicalVolume* DetectorConstruction::DefineVolumes(){
 // Geometry parameters
 G4double worldSizeX = 100.*cm;
 G4double worldSizeY = 100.*cm;
 G4double worldSizeZ  = 300.*cm;
 //
 // World Volume
 //
 G4VSolid* worldS
   = new G4Box("WorldS", worldSizeX/2, worldSizeY/2, worldSizeZ/2);
 //        ( name, Half size X, Half size Y, Half size Z)

 G4LogicalVolume* worldLV
   = new G4LogicalVolume(worldS, G4Material::GetMaterial("G4_AIR"),"WorldLV");
 // ( solid,          Material,          name)

 G4VPhysicalVolume* worldPV
   = new G4PVPlacement(nullptr,        // no rotation
                G4ThreeVector(),       // at (0,0,0)
                worldLV,               // its logical volume
                "WorldPV",             // its name
                0,                     // its mother  volume
                false,                 // no boolean operation
                0,                     // copy number
                fCheckOverlaps);       // checking overlaps
 // Visualization attributes
 //worldLV->SetVisAttributes (G4VisAttributes::GetInvisible());
```

The WorldVolume

Solid

LogicalVolume

PhysicalVolume (Placement)

Ex2 Create a cylindrical monitor (Dummy monitor) (0)

- Edit the code of DetectorConstruction.cc

```
G4VPhysicalVolume* DetectorConstruction::DefineVolumes(){
 // Geometry parameters
 G4double worldSizeX = 100.*cm;
 G4double worldSizeY = 100.*cm;
 G4double worldSizeZ  = 300.*cm;
 //
 // World Volume
 //
 G4VSolid* worldS
   = new G4Box("WorldS", worldSizeX/2, worldSizeY/2, worldSizeZ/2);
 //        ( name, Half size X, Half size Y, Half size Z)

 G4LogicalVolume* worldLV
   = new G4LogicalVolume(worldS, G4Material::GetMaterial("G4_AIR"),"WorldLV");
 // ( solid,          Material,           name)

 G4VPhysicalVolume* worldPV
   = new G4PVPlacement(nullptr,        // no rotation
                G4ThreeVector(),       // at (0,0,0)
                worldLV,               // its logical volume
                "WorldPV",             // its name
                0,                     // its mother  volume
                false,                 // no boolean operation
                0,                     // copy number
                fCheckOverlaps);       // checking overlaps
 // Visualization attributes
 //worldLV->SetVisAttributes (G4VisAttributes::GetInvisible());
```

If you have closed the vscode window,
$ code ~/Galet-v11-MedEx-01/src/DetecrtorConstruction.cc

The structure of code in the WorldVolume is common for your geometry description.
e.g. Solid,
LogicalVolume,
PhysicalVolume (Placement)

Let's copy these colored lines,
and past it to the following space.
Then, modify the variable name etc.
(See following slides)

Ex2 Create a cylindrical monitor (Dummy monitor) (0)

```
//           ( name, Half size X, Half size Y, Half size Z)

G4LogicalVolume* worldLV
  = new G4LogicalVolume(worldS, G4Material::GetMaterial("G4_AIR"),"WorldLV");
//                    ( solid,           Material,               name)

G4VPhysicalVolume* worldPV
  = new G4PVPlacement(nullptr,          // no rotation
    G4ThreeVector(),  // at (0,0,0)
    worldLV,          // its logical volume
    "WorldPV",        // its name
    0,                // its mother  volume
    false,            // no boolean operation
    0,                // copy number
    fCheckOverlaps);  // checking overlaps
// Visualization attributes
//worldLV->SetVisAttributes (G4VisAttributes::GetInvisible());

//
// User's geometries in the world volume are here
//


//
```

Insert your geometry description here

(2) Past it here

```
//....oooOOOOOooo.......oooOOOOOooo.......oooOOOOOooo.......oooOOOOOooo.....
G4VPhysicalVolume* DetectorConstruction::DefineVolumes(){
  // Geometry parameters
  G4double worldSizeX = 100.*cm;
  G4double worldSizeY = 100.*cm;
  G4double worldSizeZ  = 300.*cm;
  //
  // World Volume
  //
  G4VSolid* worldS
    = new G4Box("WorldS", worldSizeX/2, worldSizeY/2, worldSizeZ/2);
  //           ( name, Half size X, Half size Y, Half size Z)

  G4LogicalVolume* worldLV
    = new G4LogicalVolume(worldS, G4Material::GetMaterial("G4_AIR"),"WorldLV");
  //                    ( solid,           Material,               name)

  G4VPhysicalVolume* worldPV
    = new G4PVPlacement(nullptr,          // no rotation
      G4ThreeVector(),  // at (0,0,0)
      worldLV,          // its logical volume
      "WorldPV",        // its name
      0,                // its mother  volume
      false,            // no boolean operation
      0,                // copy number
      fCheckOverlaps);  // checking overlaps
  // Visualization attributes
  //worldLV->SetVisAttributes (G4VisAttributes::GetInvisible());

  //
  // User's geometries in the world volume are here
  //
```

(1) Copy these lines

Ex2 Edit code for an Al frame in the WorldVolume

```
… snipped (WorldVolume description)
// Visualization attributes
 //worldLV->SetVisAttributes (G4VisAttributes::GetInvisible());

 //
 // User's geometries in the world volume are here
 //
 G4VSolid* FrameS
  = new G4Tubs("FrameS", 0.*mm, 145.*mm, 30.*mm, 0., twopi);
 //         ( name,    Rin,   Rout,   Half size Z, sPhi, dPhi)

 G4LogicalVolume* FrameLV
  = new G4LogicalVolume(FrameS, G4Material::GetMaterial("G4_Al"),"FrameLV");
 //                 ( solid,          Material,          name)

 G4VPhysicalVolume* FramePV
  = new G4PVPlacement(nullptr,        // no rotation
                 G4ThreeVector(0, 0, 1450.*mm),  // at (0,0,0)
                 FrameLV,        // its logical volume
                 "FramePV",       // its name
                 worldLV,             // its mother  volume
                 false,          // no boolean operation
                 0,              // copy number
                 fCheckOverlaps);  // checking overlaps
G4VisAttributes* FrameVisAtt = new G4VisAttributes(G4Colour(0.0,0.0,1.0,0.3));
FrameVisAtt->SetVisibility(true);
FrameLV->SetVisAttributes(FrameVisAtt);
```

Modify these red parts.
The blue parts were for visualization parameters
G4Colour (R, G, B, Opacity);

(Note) The mother volume is the WorldVolume

Ex2 Edit code for an Air chamber in the Al frame

```
G4VSolid* ChamberS
  = new G4Tubs("ChamberS", 0.*mm, 145.*mm, 29.96*mm, 0., twopi);
//         ( name,    Rin,   Rout,   Half size Z, sPhi, dPhi)

 G4LogicalVolume* ChamberLV
  = new G4LogicalVolume(ChamberS, G4Material::GetMaterial("G4_AIR"),"ChamberLV");
//                 ( solid,          Material,           name)

 G4VPhysicalVolume* ChamberPV
  = new G4PVPlacement(nullptr,        // no rotation
                 G4ThreeVector(0, 0, 0.*mm),  // at (0,0,0)
                 ChamberLV,         // its logical volume
                 "ChamberPV",       // its name
                 FrameLV,           // its mother  volume
                 false,          // no boolean operation
                 0,              // copy number
                 fCheckOverlaps);  // checking overlaps
G4VisAttributes* ChamberVisAtt = new G4VisAttributes(G4Colour(0.0,0.5,0.5,0.3));
ChamberVisAtt->SetVisibility(true);
ChamberLV->SetVisAttributes(ChamberVisAtt);
```

(Note) The mother volume is the Al Frame

Copy lines of the Al frame description, and past it the space following the Al frame.
Then, modify the red parts.

Ex2 Edit code for an Electrode in the Air Chamber

```
…snipped ( Air Chamber description )

 G4VSolid* ElecS
  = new G4Tubs("ElecS", 0.*mm, 125.*mm, 0.0075*mm, 0., twopi);
//         ( name,    Rin,   Rout,   Half size Z, sPhi, dPhi)

 G4LogicalVolume* ElecLV
  = new G4LogicalVolume(ElecS, G4Material::GetMaterial("G4_Al"),"ChamberLV");
//                ( solid,         Material,           name)

 G4VPhysicalVolume* ElecPV0
  = new G4PVPlacement(nullptr,        // no rotation
                 G4ThreeVector(0, 0, -10.*mm),  // at (0,0,0)
                 ElecLV,         // its logical volume
                 "ElecPV0",       // its name
                 ChamberLV,              // its mother  volume
                 false,          // no boolean operation
                 0,              // copy number
                 fCheckOverlaps);  // checking overlaps
```

Copy lines of the Air chamber description, and past it the space following the Air chamber.
Then, modify the red parts.

(Note) The mother volume is the Air chamber

These codes place an electrode in the air chamber. We need to place one more electrode.

Ex2 Edit code for an Electrode in the Air Chamber

```
G4VSolid* ElecS
  = new G4Tubs("ElecS", 0.*mm, 125.*mm, 0.0075*mm, 0., twopi);
//         ( name,    Rin,   Rout,   Half size Z, sPhi, dPhi)

 G4LogicalVolume* ElecLV
  = new G4LogicalVolume(ElecS, G4Material::GetMaterial("G4_Al"),"ChamberLV");
//                ( solid,         Material,           name)

 G4VPhysicalVolume* ElecPV0
  = new G4PVPlacement(nullptr,        // no rotation
                G4ThreeVector(0, 0, -10.*mm),  // at (0,0,0)
                ElecLV,         // its logical volume
                "ElecPV0",       // its name
                ChamberLV,             // its mother  volume
                false,          // no boolean operation
                0,              // copy number
                fCheckOverlaps);  // checking overlaps

 G4VPhysicalVolume* ElecPV1
  = new G4PVPlacement(nullptr,        // no rotation
                G4ThreeVector(0, 0, +10.*mm),  // at (0,0,0)
                ElecLV,         // its logical volume
                "ElecPV1",       // its name
                ChamberLV,             // its mother  volume
                false,          // no boolean operation
                1,              // copy number
                fCheckOverlaps);  // checking overlaps
```

Add codes for one more electrode.

Add the red parts, which place the same logical-volume (ElecLV) in the ChamberLV with the different position (0, 0, +10*mm). The copy number of 1.

(Note) Name ElecPV1

(Note) The position (0, 0, +10*mm)

(Note) Place ElecLV

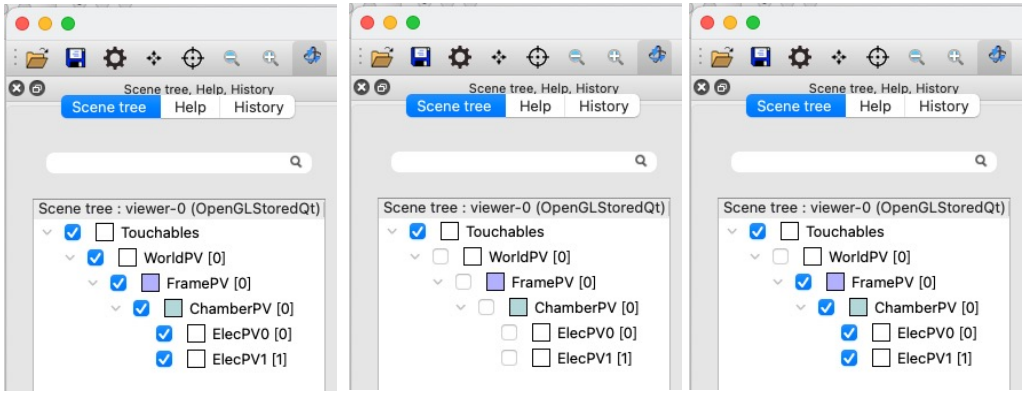
(Note) in the ChamberLV

(Note) The copy number is to be 1

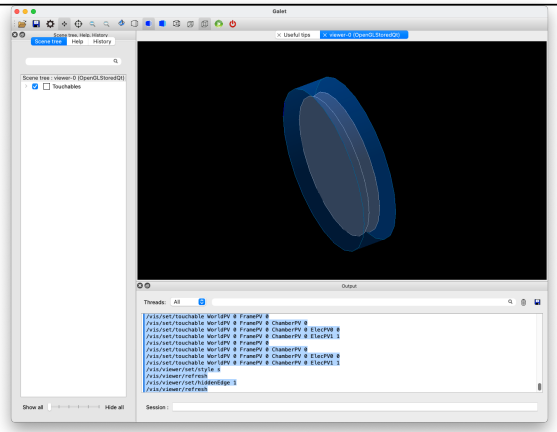
Ex2 Save the code and rebuild the application

- Save the DectorConstruction.cc
- $ make
- $ ./Galet

Check the structure in “Scene tree”



Session: /vis/drawVolume FramePV



Ex3 An example of water phantom (GaletPhantom class)

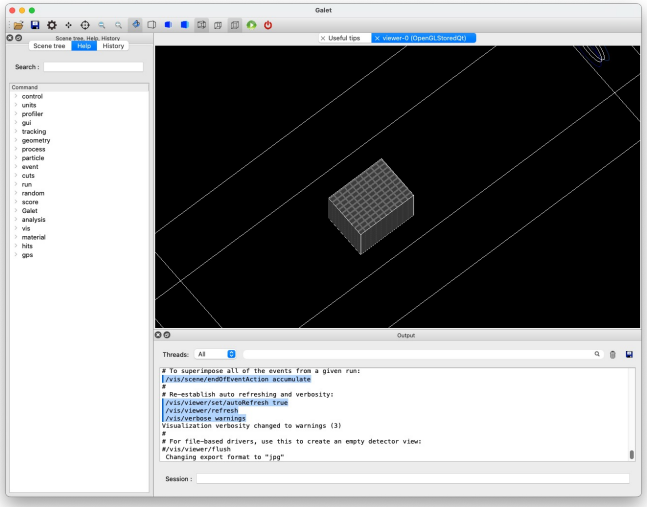
- Voxelated geometry using the 3D replica and parameterization.
  - GaletPhantom
  - GaletNestedPhantomParameterisation
- The inside of the code is not simple, so we skip the detail explanation of the code.
- Let's utilize this class as a water phantom.

Snippet from GaletPhantom.cc

```
G4LogicalVolume* GaletPhantom::ConstructPhantom(){

  G4String phantomNameS("PhantomS");
  G4VSolid* phantomS = new G4Box(phantomNameS,
                                 fNoVoxelX*fVoxelSPCX/2.,
                                 fNoVoxelY*fVoxelSPCY/2.,
                                 fNoVoxelZ*fVoxelSPCZ/2.);
  G4String phantomNameLV("PhantomLV");
  fPhantomLV = new G4LogicalVolume(phantomS,fAir,phantomNameLV);

  //----- Replication of Water Phantom Volume.
  //--- Y Slice
  G4String yRepName("RepY");
  G4VSolid* solYRep = new G4Box(yRepName,fNoVoxelX*fVoxelSPCX/2.,
                                fVoxelSPCY/2.,
                                fNoVoxelZ*fVoxelSPCZ/2.);
  G4LogicalVolume* logYRep = new G4LogicalVolume(solYRep,fAir,yRepName);
  new G4PVReplica(yRepName,logYRep,fPhantomLV,kYAxis,
                  fNoVoxelY,fVoxelSPCY);

  logYRep->SetVisAttributes(new G4VisAttributes(G4VisAttributes::GetInvisible()));

  //--- X Slice
  G4String xRepName("RepX");
  G4VSolid* solXRep = new G4Box(xRepName,fVoxelSPCX/2.,fVoxelSPCY/2.,
                                fNoVoxelZ*fVoxelSPCZ/2.);
  G4LogicalVolume* logXRep = new G4LogicalVolume(solXRep,fAir,xRepName);
  new G4PVReplica(xRepName,logXRep,logYRep,kXAxis,fNoVoxelX,fVoxelSPCX);

  logXRep->SetVisAttributes(new G4VisAttributes(G4VisAttributes::GetInvisible()));

  //----- Voxel solid and logical volumes
  //--- Z Slice
  G4VSolid* solVoxel = new G4Box("voxelS",fVoxelSPCX/2.,
                                 fVoxelSPCY/2.,fVoxelSPCZ/2.);

  G4LogicalVolume* logicVoxel =
    new G4LogicalVolume(solVoxel,fAir,fVoxelLVName);
  if (!fVisOn){
    logicVoxel->
      SetVisAttributes(new G4VisAttributes(G4VisAttributes::GetInvisible()));
  }

  //logicVoxel->
  //SetVisAttributes(new G4VisAttributes(G4VisAttributes::GetInvisible()));

  //
  // Parameterisation for transformation of voxels.
  //  (voxel size is fixed in this example.
  //    e.g. nested parameterisation handles material
  //    and transfomation of voxels.)
  G4ThreeVector voxelSize(fVoxelSPCX/2.,fVoxelSPCY/2.,fVoxelSPCZ/2.);
  GaletNestedPhantomParameterisation* param =
    new GaletNestedPhantomParameterisation(voxelSize,
                                           fMaterialList,
                                           fMaterialIndices,
                                           fNoVoxelX,fNoVoxelY,fNoVoxelZ);
  param->SetVisAttributes(fVisAtt);
  param->SetVisOn(fVisOn);

  new G4PVParameterised("phantom",    // their name
                        logicVoxel, // their logical volume
                        logXRep,      // Mother logical volume
                        kZAxis,       // Are placed along this axis
                        //kUndefined,
                        // Are placed along this axis
                        fNoVoxelZ,      // Number of cells
                        param);       // Parameterisation.

  // Score volume.
  fScoreVoxelLV = logicVoxel;

  return fPhantomLV;
}
```

Ex3 An example of water phantom (GaletPhantom class)

- In the DetectorConstruction.cc, the lines for the water-phantom are commented out.
  - Let's activate those lines
  - Rebuild the application and Run



Remove /* and */

```
… snipped (Electrodes)
G4VPhysicalVolume* ElecPV1
  = new G4PVPlacement(nullptr,        // no rotation
                      G4ThreeVector(0, 0, +10.*mm),  // at (0,0,0)
                      ElecLV,         // its logical volume
                      "ElecPV1",      // its name
                      ChamberLV,            // its mother  volume
                      false,          // no boolean operation
                      1,              // copy number
                      fCheckOverlaps);  // checking overlaps
 //
 // WaterPhantom
  /*
 GaletPhantom* phantom = new GaletPhantom("phantomVoxelLV");
 phantom->SetWaterPhantom(15.*cm,15.*cm,20.*cm,10,10,10);
 //                                dxHalf,dyHalf,dzHalf,Nx,Ny,Nz
 phantom->SetVisOn(true);
 G4LogicalVolume* phantomLV = phantom->ConstructPhantom();
 G4RotationMatrix* rotation = new G4RotationMatrix();
 rotation->rotateY(180.*degree);
 new G4PVPlacement(rotation, G4ThreeVector(0,0,-20.0*cm),
                   phantomLV, "PhantomLV",
                   worldLV, false, 0, fCheckOverlaps);
  */
 //
 // Always return the physical World
 //
 return worldPV;
}
```