

```
////////////////////////////////////
/*      LUXSimExampleSimpleDetector.cc
*
* This is the code file to define the LZ20 Detector which is composed of a
* water shield, scintillator (which for now uses the properties of water),
* and liquid Xenon encased in Titanium.
*
*****
* Change log
*      26 March 2010 - Initial submission (Monica)
*      26 March 2010 - altered to take out warnings when compiling
*/
////////////////////////////////////

//
//      C/C++ includes
//
#include <sstream>

//
//      GEANT4 includes
//
#include "globals.hh"
#include "G4LogicalVolume.hh"
#include "G4Tubs.hh"

//
//      LUXSim includes
//
#include "LUXSimLZ20Simple.hh"
#include "LUXSimDetectorComponent.hh"
// #include "LUXSimExamplePMTBank.hh"
#include "LUXSimMaterials.hh"

using namespace std;

//-----++++++-----++++++-----++++++-----++++++-----++++++-----
//                                     LUXSimLZ20Simple
//-----++++++-----++++++-----++++++-----++++++-----++++++-----
LUXSimLZ20Simple::LUXSimLZ20Simple()
{
    //      Get the LUXSimMaterials pointer
    LUXSimMaterials *luxMaterials = LUXSimMaterials::GetMaterials();

    // world volume *****

    G4double worldRadius = 11.0*m;
    G4double worldHeight = 22.0*m;

    G4Tubs* world_tube =
new G4Tubs(
    "world_tube",
    0.*cm,
    worldRadius,
    0.5*worldHeight,
```

```
        0.*deg,
            360.*deg
    );
    logicalVolume =
new G4LogicalVolume(
    world_tube,
    luxMaterials->Vacuum(),
    "world_log"
);
logicalVolume->SetVisAttributes(luxMaterials->VacuumVis());

//      The water shield
G4double vacuumVesselRadius = 6.*m;
    //outer Radius, inner radius will be outer radius of vacuum
G4double vacuumVesselHeight = 12.*m;
//G4double vacuumVesselThickness = 4.995*m;
G4Tubs *water_solid = new G4Tubs( "water_solid", 0.*cm,
    vacuumVesselRadius, vacuumVesselHeight/2.,0.*deg,360.*deg);
G4LogicalVolume *water_log = new G4LogicalVolume (water_solid,
    luxMaterials->Water(), "water_log" );
water_log->SetVisAttributes( luxMaterials->WaterVis() );

LUXSimDetectorComponent *water_phys;
water_phys= new LUXSimDetectorComponent( 0, G4ThreeVector(),
    water_log, "Water", logicalVolume, 0, 0, false );

//Dimensions of scintillator, Titanium, and the liquid Xenon cavity
//
//
G4double cavityDiameter = 2.*m;
G4double cavityHeight = 2.*m;
G4double ti_thickness = 3.*mm;
G4double scint_thickness = 0.8*m;

// Scintillator which has the same properties as water
G4Tubs *scint_solid = new G4Tubs( "scint_solid", 0*mm,
    cavityDiameter/2 + scint_thickness, cavityHeight/2 + scint_thickness,
    0.*deg, 360.*deg );
G4LogicalVolume *scint_log = new G4LogicalVolume( scint_solid,
    luxMaterials->Water(), "scint_log" );
scint_log->SetVisAttributes( luxMaterials->CopperVis() );

LUXSimDetectorComponent *scint_phys;
scint_phys = new LUXSimDetectorComponent( 0, G4ThreeVector(),
    scint_log, "Scintillator", water_log, 0, 0, false );

//Titanium layer
G4Tubs *cryostat_solid = new G4Tubs( "cryostat_solid", 0*mm,
    cavityDiameter/2 + ti_thickness, cavityHeight/2 + ti_thickness,
    0.*deg, 360.*deg );
G4LogicalVolume *cryostat_log = new G4LogicalVolume( cryostat_solid,
    luxMaterials->Titanium(), "cryostat_log" );
cryostat_log->SetVisAttributes( luxMaterials->TitaniumVis() );
```

```
LUXSimDetectorComponent *cryostat_phys;
cryostat_phys = new LUXSimDetectorComponent( 0, G4ThreeVector(),
                                              cryostat_log, "Titanium", scint_log, 0, 0, false );

//liquid Xenon
G4Tubs *liquidXeTarget_solid = new G4Tubs( "liquidXeTarget_solid", 0*mm,
                                             cavityDiameter/2, cavityHeight/2, 0.*deg,360.*deg );
G4LogicalVolume *liquidXeTarget_log = new G4LogicalVolume(
    liquidXeTarget_solid, luxMaterials->LiquidXe(),
    "liquidXeTarget_log" );
liquidXeTarget_log->SetVisAttributes( luxMaterials->LiquidXeVis() );
liquidXeTarget = new LUXSimDetectorComponent( 0, G4ThreeVector(),
                                              liquidXeTarget_log, "LiquidXeTarget", cryostat_log, 0, 0, false );

}

//-----++++++-----++++++-----++++++-----++++++-----++++++-----++++++-----
//                                     ~LUXSimLZ20Simple
//-----++++++-----++++++-----++++++-----++++++-----++++++-----++++++-----
LUXSimLZ20Simple::~LUXSimLZ20Simple() {}
```