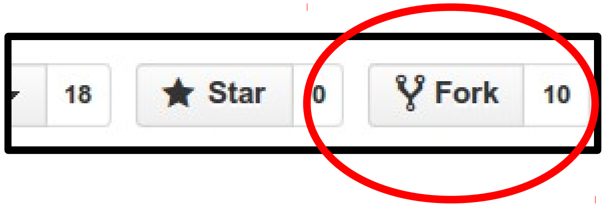
WCSim and νPRISM

Mark Scott for the νPRISM collaboration
νPRISM analysis meeting
15th March 2015 - IPMU

νPRISM TRIUMF

First things first...

- Need to download some software to really look at WCSim
- First, fork the Analysis, WCSim and temp_event_display repositories in the nuPRISM github organisation (https://github.com/nuPRISM) to your personal github account
  - On the page linked above click on a repository
  - Click on the 'Fork' button at the top right
  - Select your account
  - If you already forked WCSim from the WCSim organisation you will have to remove that repository from your account (saving any code you are working on) then fork the nuPRISM version



- Now, git clone the Analysis repository to your computer:
  - 'git clone https://github.com/your_username/Analysis.git'
  - Or 'git clone git@github.com:your_username/Analysis.git' if you have ssh keys setup

Installing everything

- WCSim doesn't get on well with gcc versions 4.6 and higher, it works with 4.4.7 and 4.3
  - Run 'gcc --version' to check which version of gcc you have
  - Install an older version of gcc if necessary and point the /usr/bin/g++ and /usr/bin/gcc symlinks to /usr/bin/g++-4.4 and /usr/bin/gcc-4.4

- You should have a local copy of the Analysis repository, cd into it
- Set the 'GITNAME' environment variable to your git username
- Now do:

```
source Source_At_Start_nuPRISM.sh

source nuPRISM_Install.sh build
```

- This should start downloading and building (almost) everything we need – this takes about 30 minutes, so get it going now

Installing DAWN

- The event display is alright, but does NOT display the GEANT4 geometry produced by WCSim
- We need a GEANT4 visualiser to do that – the only one I've had any luck with is called DAWN
- You don't have to do this – the visualiser is not very user friendly – but feel free to if you want to modify any WCSim geometry
- http://ific.uv.es/~silicio/simulation/athena_installation_dawn.htm Shows you how to download and compile everything
  - I used 'sudo apt-get install gv' to get ghostview, but you just need a programme that can draw postscript files
  - DAWN worked fine on SL6, on Ubuntu I had to modify it quite a lot – you can download my modified version here http://trshare.triumf.ca/~mscott/dawn.tgz

Last thing!

- We want to run WCSim on some beam neutrinos, so download this NEUT file
  http://trshare.triumf.ca/~mscott/genev_320a_1km_nd3_9xx_30818.root

- This has some neutrino interactions in the middle section of nuPRISM-lite

- To check WCSim built correctly, you should have the WCSim executable in this directory:

```
WCSim/exe/bin/Linux-g++/WCSim
```

νPRISM

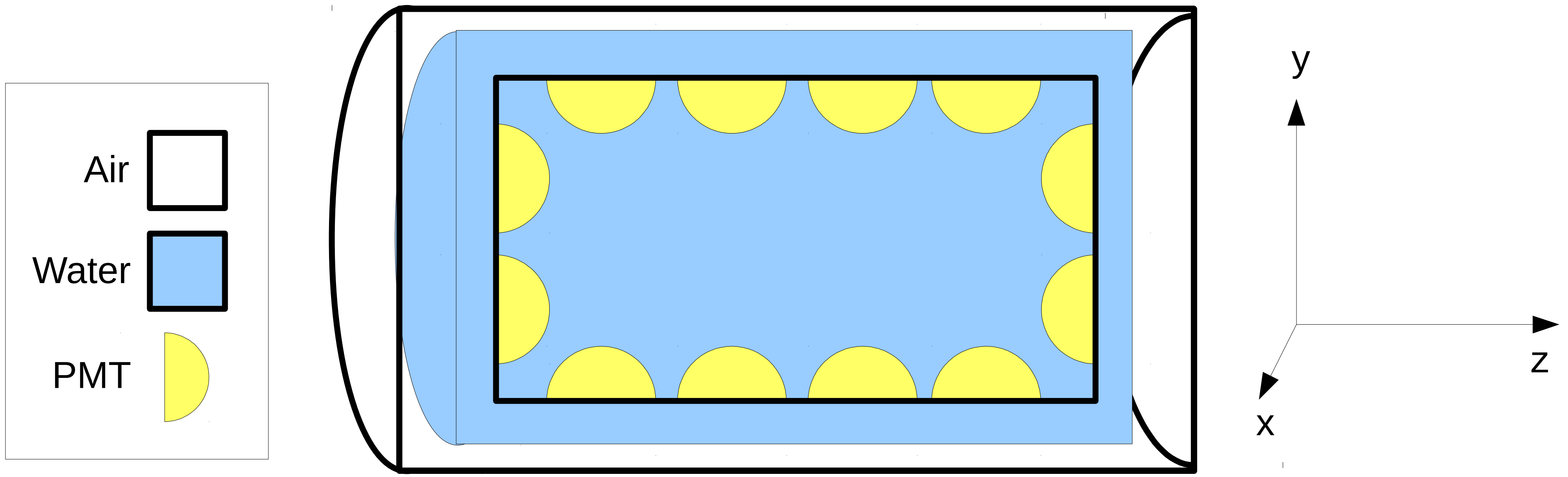
What is WCSim?

- **W**ater **C**herenkov **Sim**ulator – simulates water cherenkov detectors
- GEANT4 based
- Will be the default HyperK MC – lots of work starting on it

- What does it do:
  - Simulates cylinders and specific detector designs, such as the original HK egg-shaped tank
  - Propagates particles through detector
  - Simulates cherenkov light production
  - Simulates PMT response to cherenkov light
  - Produces an output file with the PMT response, true particle information and detector geometry information

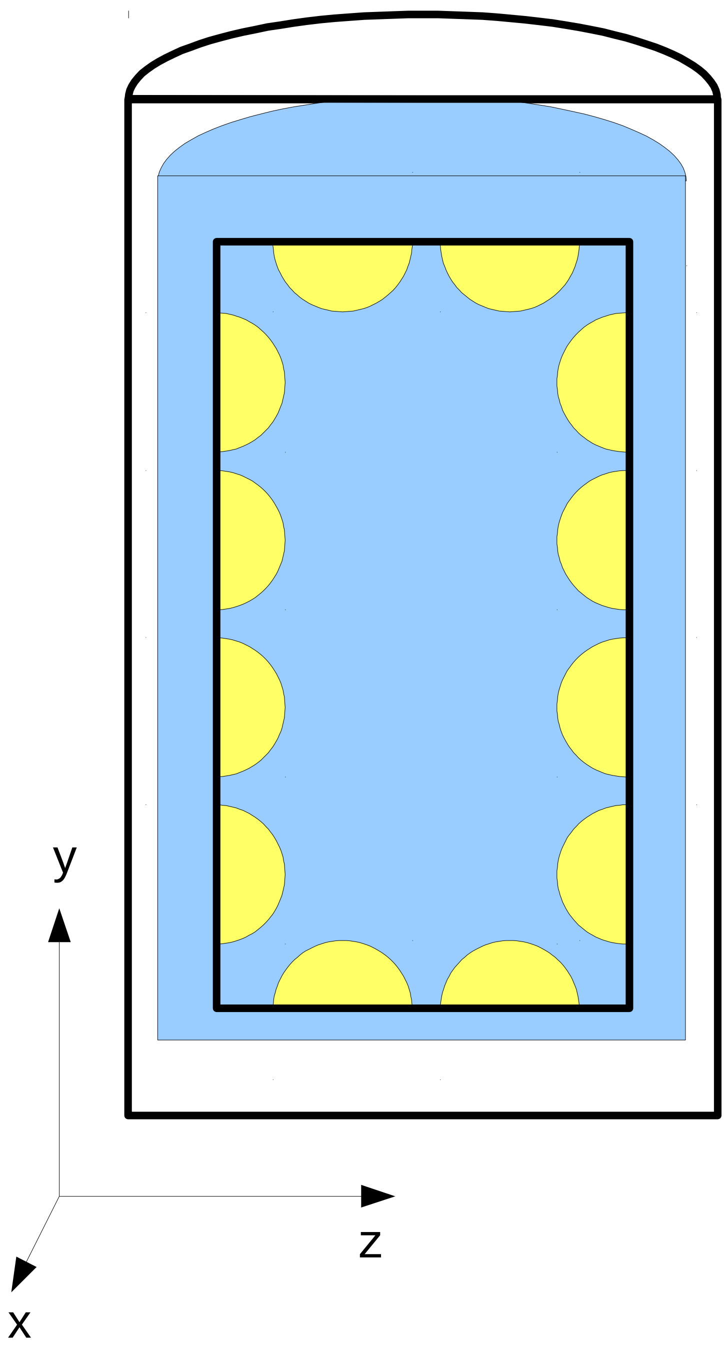
Cylinders in WCSim

- Use standard cylinder construction to make nuPRISM
- Outer cylinder filled with air
- Smaller inner cylinder filled with water
- Still smaller instrumented cylinder within water cylinder
- By default, in WCSim, cylinder orientated with long axis along global z-axis



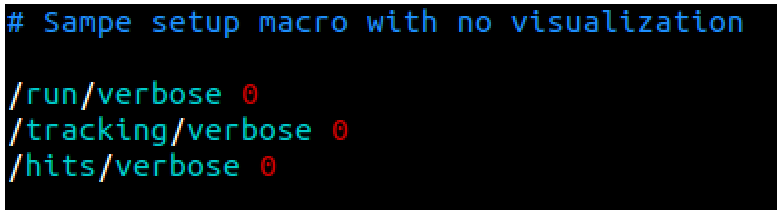
nuPRISM in WCSim

- Use standard cylinder construction to make nuPRISM
- Keep same nested cylinders
- Rotate to align WCSim Z-axis with beam coordinate system z-axis (same as ND280)
- Cylinder axis aligned with WCSim y-axis
- Can set y-axis position of cylinder
  - In effect sets off-axis angle of detector
- Uses same coordinates as NEUT files produced by Mark H.
  - Removes 1km Z offset
  - User must match Y position of detector to neutrino production plane position



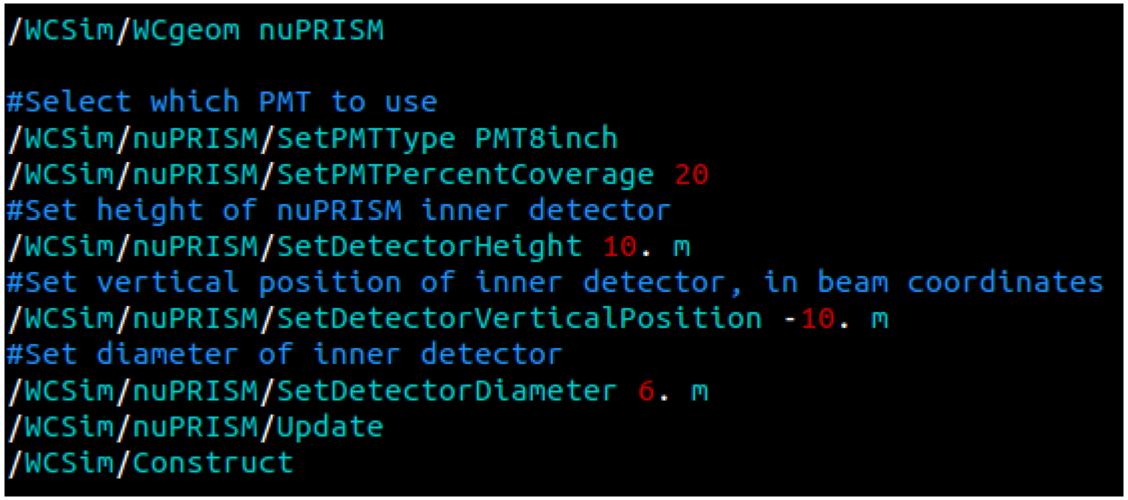
Controlling WCSim

- WCSim is controlled by macro files – novis.mac and vis.mac in the WCSim directory
- Let's look at novis.mac – stands for 'no visualisation'
- First, sets the verbosity to 0



```
# Sampe setup macro with no visualization

/run/verbose 0
/tracking/verbose 0
/hits/verbose 0
```

- Next, some commands to build the geometry – ignore for now
- /WCSim/WCgeom selects which geometry we want
- SetDetectorVerticalPosition – this is in the beam coordinate system, so 2.5 degrees off-axis is around -20m
- /WCSim/nuPRISM/Update loads the previous options



```
/WCSim/WCgeom nuPRISM

#Select which PMT to use
/WCSim/nuPRISM/SetPMTType PMT8inch
/WCSim/nuPRISM/SetPMTPercentCoverage 20
#Set height of nuPRISM inner detector
/WCSim/nuPRISM/SetDetectorHeight 10. m
#Set vertical position of inner detector, in beam coordinates
/WCSim/nuPRISM/SetDetectorVerticalPosition -10. m
#Set diameter of inner detector
/WCSim/nuPRISM/SetDetectorDiameter 6. m
/WCSim/nuPRISM/Update
/WCSim/Construct
```

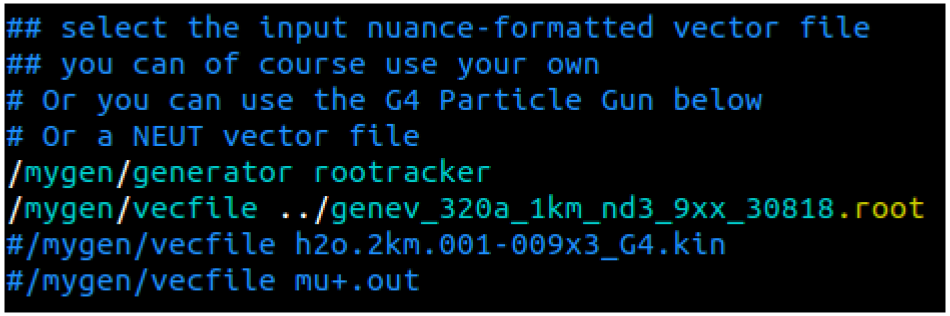
- /WCSim/Construct builds the detector

νPRISM

Controlling WCSim - 2

TRIUMF

- Next we have some PMT simulation methods and an option to save pi0 truth info, will ignore for now
- Now we have the interaction generator
- Options are:



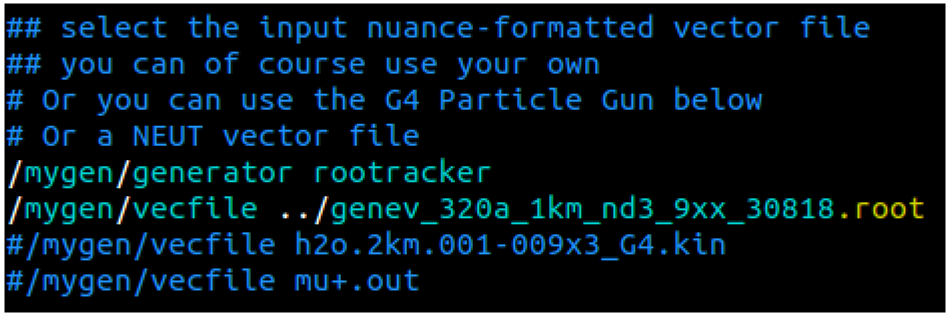
```
## select the input nuance-formatted vector file
## you can of course use your own
# Or you can use the G4 Particle Gun below
# Or a NEUT vector file
/mygen/generator rootracker
/mygen/vecfile ../genev_320a_1km_nd3_9xx_30818.root
#/mygen/vecfile h2o.2km.001-009x3_G4.kin
#/mygen/vecfile mu+.out
```

  - Normal - particle guns
  - Muline – nuance formatted text file as input
  - Laser – simulate a laser
  - Rootracker – use RooTracker formatted input, such as NEUT

- /mygen/vecfile points to the input particle vector file, if you are using the 'muline' or 'rootracker' generators

Controlling WCSim - 2

- Next we have some PMT simulation methods and an option to save pi0 truth info, will ignore for now
- Now we have the interaction generator

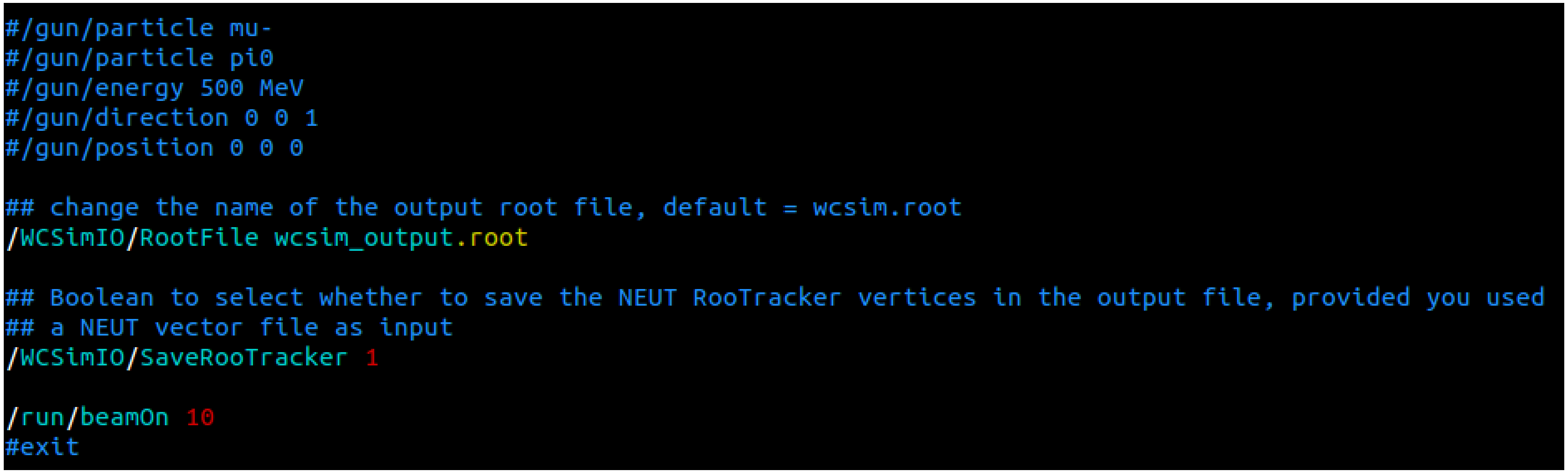


```
## select the input nuance-formatted vector file
## you can of course use your own
# Or you can use the G4 Particle Gun below
# Or a NEUT vector file
/mygen/generator rootracker
/mygen/vecfile ../genev_320a_1km_nd3_9xx_30818.root
#/mygen/vecfile h2o.2km.001-009x3_G4.kin
#/mygen/vecfile mu+.out
```

- Options are:
  - Normal - particle guns
  - Muline – nuance formatted text file as input
  - Laser – simulate a laser
  - Rootracker – use RooTracker formatted input, such as NEUT
- /mygen/vecfile points to the input particle vector file, if you are using the 'muline' or 'rootracker' generators

Controlling WCSim - 3

- After this there are some options to simulate dark noise rates – we'll stick to the PMT rates for now

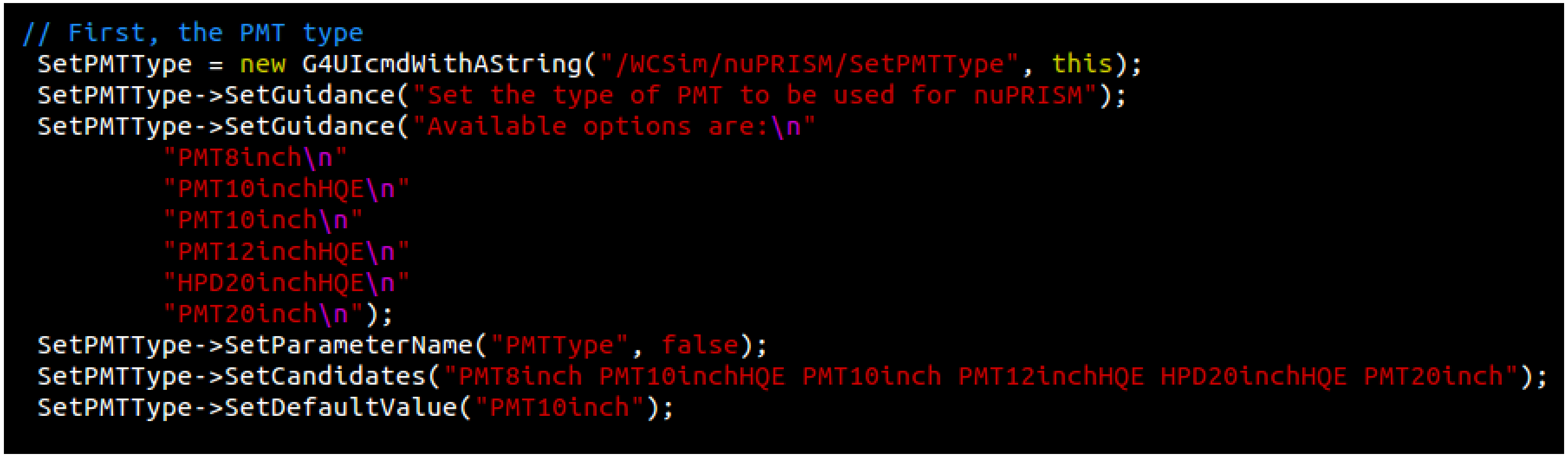


```
#/gun/particle mu-
#/gun/particle pi0
#/gun/energy 500 MeV
#/gun/direction 0 0 1
#/gun/position 0 0 0

## change the name of the output root file, default = wcsim.root
/WCSimIO/RootFile wcsim_output.root

## Boolean to select whether to save the NEUT RooTracker vertices in the output file, provided you used
## a NEUT vector file as input
/WCSimIO/SaveRooTracker 1

/run/beamOn 10
#exit
```

- /gun/ commands control particle guns
  - Position of particle gun is given in **centimetres** (blame WCSim...)
- /WCSimIO controls input and output
  - RootFile is the WCSim output file
  - SaveRooTracker will save the rootracker vertices to the output file
- /run/beamOn – how many events do you want to simulate?

WCSim Messengers

- GEANT4, and WCSim, use messenger classes to read all the options from the macros, e.g. src/WCSimDetectorMessenger.cc
- Use these to add new macro options



```
// First, the PMT type
 SetPMTType = new G4UIcmdWithAString("/WCSim/nuPRISM/SetPMTType", this);
 SetPMTType->SetGuidance("Set the type of PMT to be used for nuPRISM");
 SetPMTType->SetGuidance("Available options are:\n"
         "PMT8inch\n"
         "PMT10inchHQE\n"
         "PMT10inch\n"
         "PMT12inchHQE\n"
         "HPD20inchHQE\n"
         "PMT20inch\n");
 SetPMTType->SetParameterName("PMTType", false);
 SetPMTType->SetCandidates("PMT8inch PMT10inchHQE PMT10inch PMT12inchHQE HPD20inchHQE PMT20inch");
 SetPMTType->SetDefaultValue("PMT10inch");
```

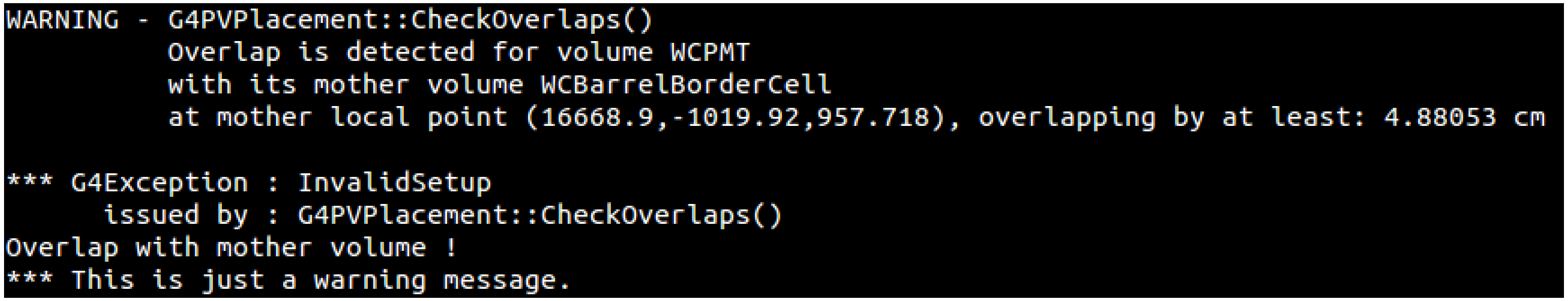
- G4UIcmdWithAXXXX – can add options with string, bool, double etc. The option name is specified as the first argument
- SetGuidance – provides some documentation
- ParameterName – unique name for the parameter filled by this option
- Candidates – list the possible values the option can take, or leave unset
- Default value – value parameter takes if the option appears in the macro file but it is not set

Running WCSim

- Current macro file should work – make sure the path to the vector file you downloaded is correct
- To run WCSim just call:

  ./exe/bin/Linux-g++/WCSim novis.mac

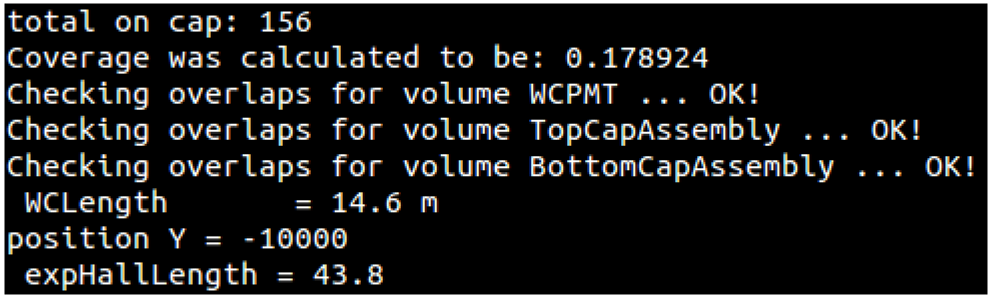
- Lots of output...
- Ignore overlap warnings – some bits of the geometry do overlap but there's nothing we can do about that



```
WARNING - G4PVPlacement::CheckOverlaps()
          Overlap is detected for volume WCPMT
          with its mother volume WCBarrelBorderCell
          at mother local point (16668.9,-1019.92,957.718), overlapping by at least: 4.88053 cm

*** G4Exception : InvalidSetup
      issued by : G4PVPlacement::CheckOverlaps()
Overlap with mother volume !
*** This is just a warning message.
```

Running WCSim - 2

- After all the 'NeutronHP:...' messages you get the bit where nuPRISM is actually built
- Shows how many PMTs fit on the caps and the PMT coverage for the caps



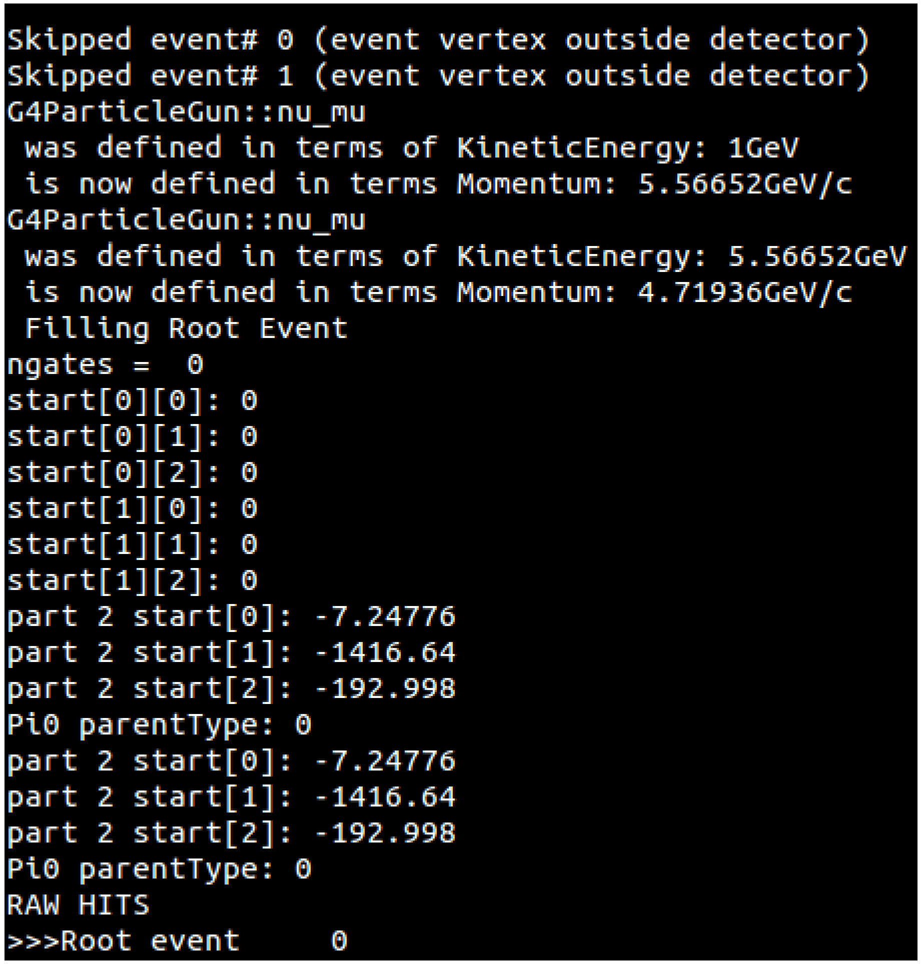
```
total on cap: 156
Coverage was calculated to be: 0.178924
Checking overlaps for volume WCPMT ... OK!
Checking overlaps for volume TopCapAssembly ... OK!
Checking overlaps for volume BottomCapAssembly ... OK!
 WCLength       = 14.6 m
position Y = -10000
 expHallLength = 43.8
```

- WCLength is the length of the air volume in Y direction – 4.6m longer than the specified inner detector
- Position Y shows the position, in mm, that the detector is being simulated at
- Next, you get a description of the models and stuff loaded in WCSim – not too interesting, but good to check if you need to

Running WCSim - 3

- Finally, we get some events!

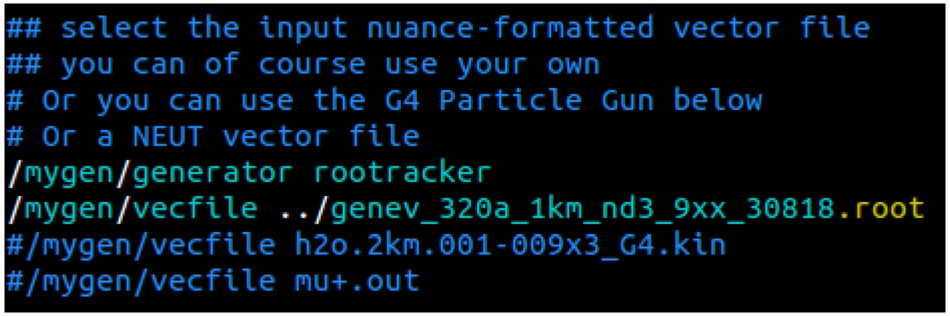
- Skipped event#
  - The NEUT files to not perfectly match the detector size or position
  - Some events happen outside the detector, so are skipped
- Kinetic energy vs momentum:
  - Not sure why these are appearing
  - Fine for neutrino – this is not used ever – but might be something we need to fix



```
Skipped event# 0 (event vertex outside detector)
Skipped event# 1 (event vertex outside detector)
G4ParticleGun::nu_mu
 was defined in terms of KineticEnergy: 1GeV
 is now defined in terms Momentum: 5.56652GeV/c
G4ParticleGun::nu_mu
 was defined in terms of KineticEnergy: 5.56652GeV
 is now defined in terms Momentum: 4.71936GeV/c
 Filling Root Event
ngates =  0
start[0][0]: 0
start[0][1]: 0
start[0][2]: 0
start[1][0]: 0
start[1][1]: 0
start[1][2]: 0
part 2 start[0]: -7.24776
part 2 start[1]: -1416.64
part 2 start[2]: -192.998
Pi0 parentType: 0
part 2 start[0]: -7.24776
part 2 start[1]: -1416.64
part 2 start[2]: -192.998
Pi0 parentType: 0
RAW HITS
>>>Root event     0
```

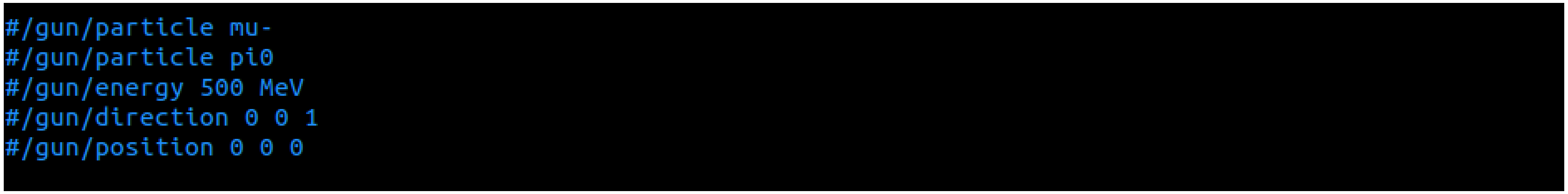
Running particle guns

- Now some particle gun
- Change the generator to 'normal'
- Comment out the vecfile option



```
## select the input nuance-formatted vector file
## you can of course use your own
# Or you can use the G4 Particle Gun below
# Or a NEUT vector file
/mygen/generator rootracker
/mygen/vecfile ../genev_320a_1km_nd3_9xx_30818.root
#/mygen/vecfile h2o.2km.001-009x3_G4.kin
#/mygen/vecfile mu+.out
```

- Uncomment all the '/gun/' options, choosing one particle type



```
#/gun/particle mu-
#/gun/particle pi0
#/gun/energy 500 MeV
#/gun/direction 0 0 1
#/gun/position 0 0 0
```

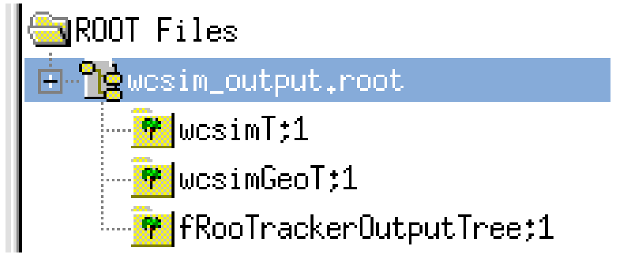
- The direction of the particle is in the nuPRISM coordinate system
- The X and Z positions (in cm) are in relation to the central axis of the cylinder – you can have negative values here
- The Y position should work with whatever you set as the VerticalPosition for the nuPRISM detector
  - Set VerticalPosition to 0 and then the Y position of the particle gun will be the vertical position about the central point of nuPRISM
- ./exe/bin/Linux_g++/WCSim novis.mac

νPRISM

Output

TRIUMF

- We get an output file
- wcsimT:
  - PMT response information
  - True particle information
- wcsimGeoT:
  - The detector geometry, in the WCSimRootGeom format (I think this should be the standard ROOT geometry format)
- Both of the above need WCSim shared object libraries to interpret, so are currently not that useable
- fRooTrackerOutputTree:
  - The RooTracker vertices used to simulate the events

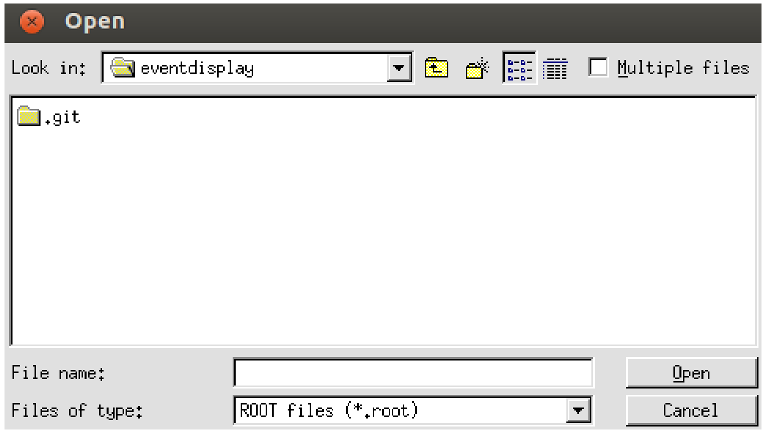

ROOT Files
wcsim_output.root
wcsimT;1
wcsimGeoT;1
fRooTrackerOutputTree;1

νPRISM

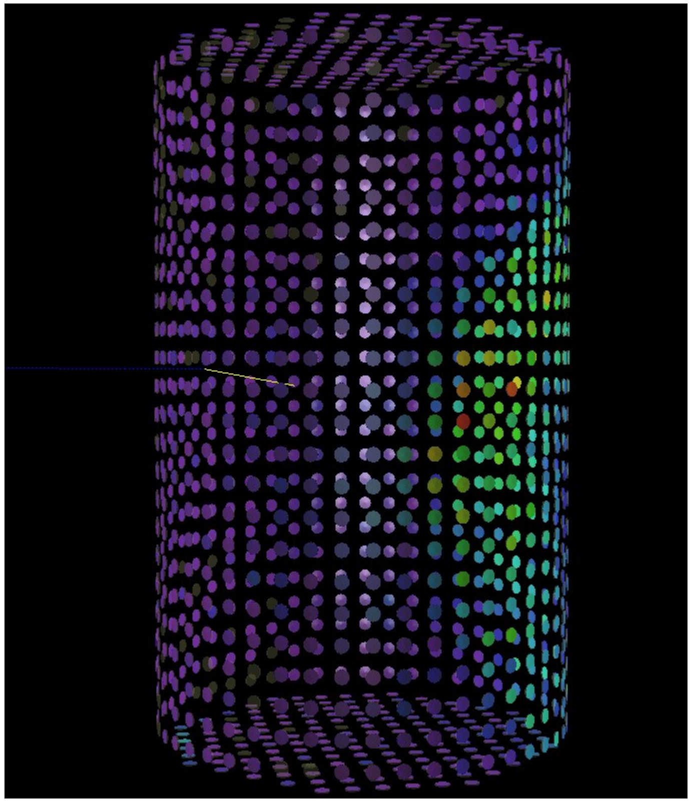
Event display

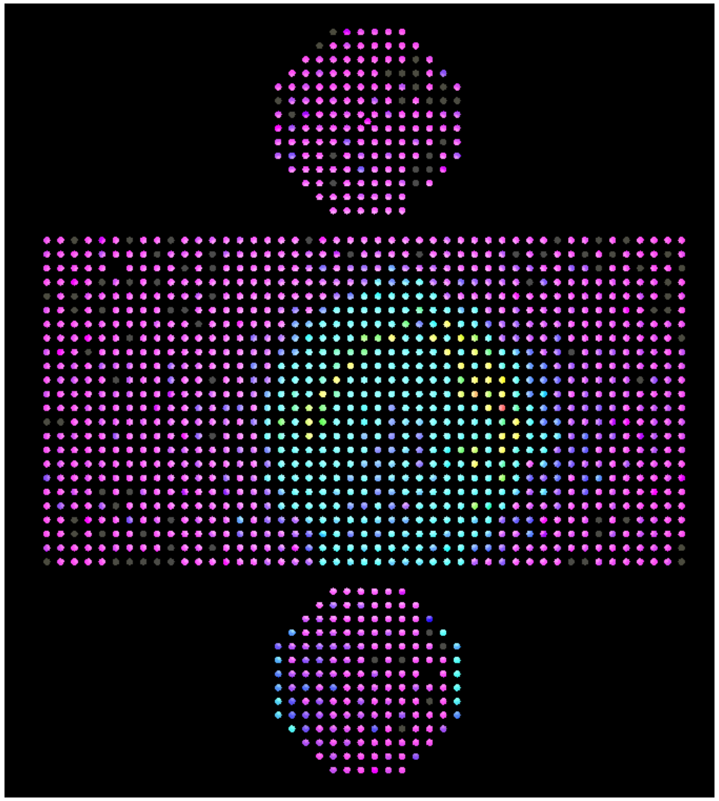
TRIUMF

- Go to the temp_event_display directory
- Should be able to run 'root -l hyperk-display.C'
- Use the window to choose your WCSim output file
- Cancel the fiTQun file



3D view

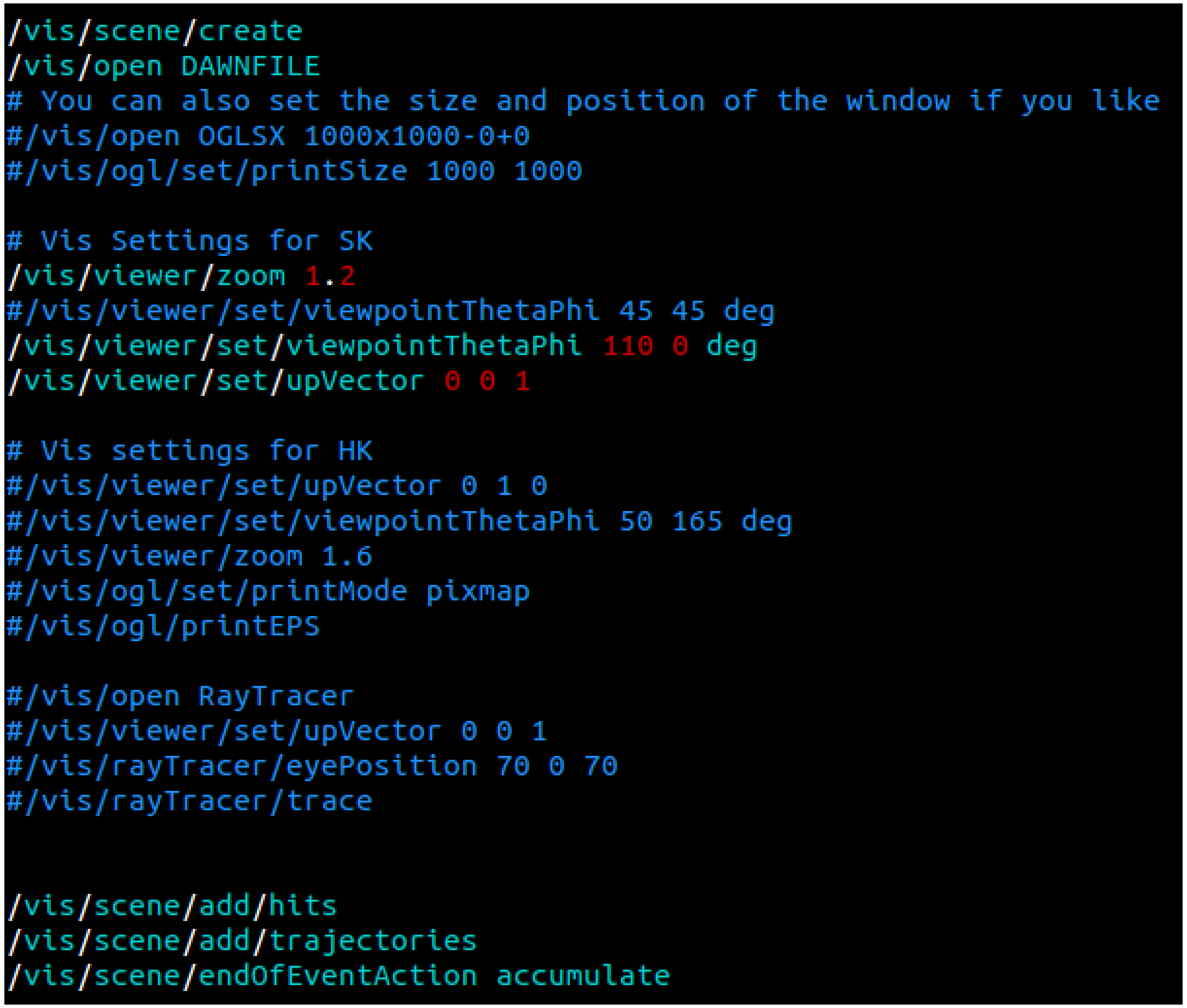




Unrolled view

Using DAWN

- Event display uses PMT position with lots of hard coded assumptions to display the detector geometry
- Sometimes you want to see what you're actually simulating
- Vis.mac file has the options on right
- /vis/open DAWNFILE makes WCSim produce g4_00.prim output file
- Other options not really needed by DAWN
- /vis/scene/add options add different G4 objects to the visualised event



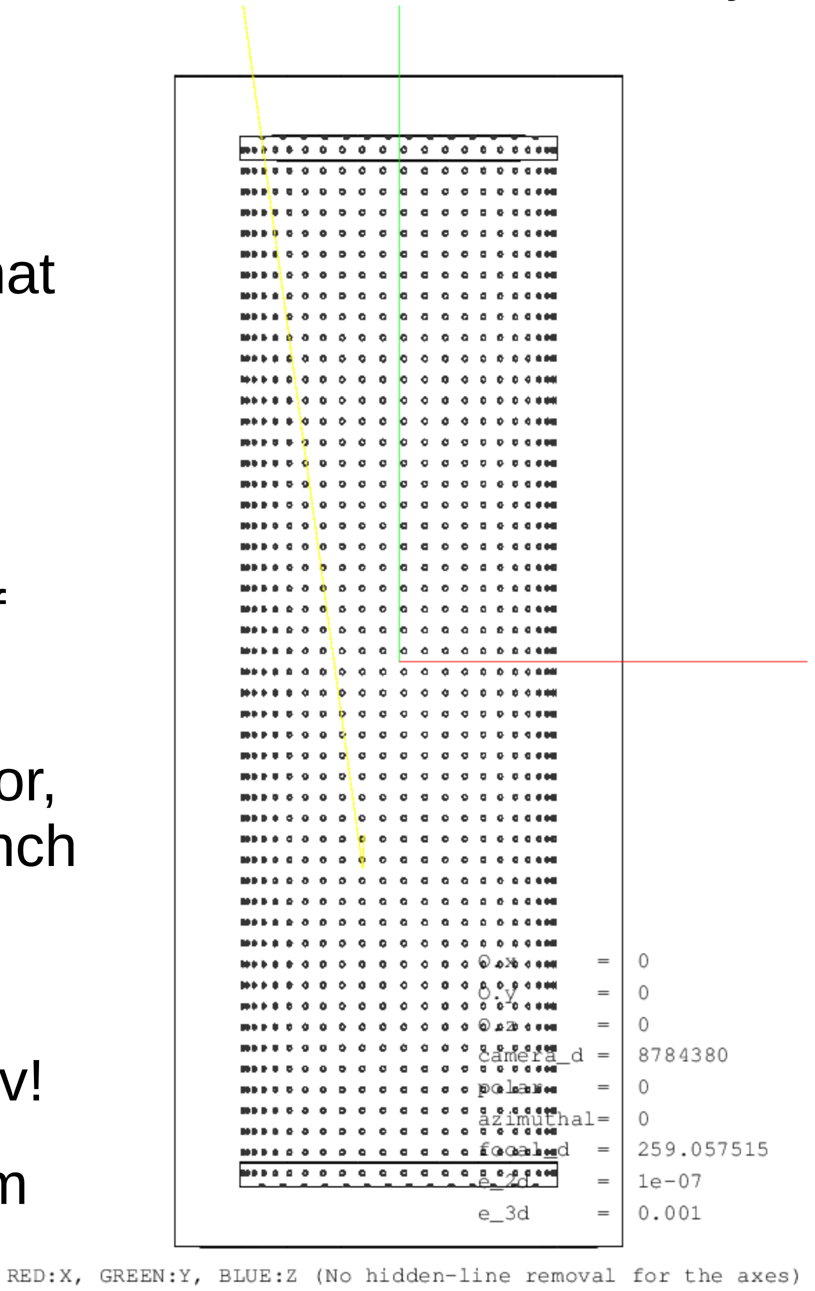
```
/vis/scene/create
/vis/open DAWNFILE
# You can also set the size and position of the window if you like
#/vis/open OGLSX 1000x1000-0+0
#/vis/ogl/set/printSize 1000 1000

# Vis Settings for SK
/vis/viewer/zoom 1.2
#/vis/viewer/set/viewpointThetaPhi 45 45 deg
/vis/viewer/set/viewpointThetaPhi 110 0 deg
/vis/viewer/set/upVector 0 0 1

# Vis settings for HK
#/vis/viewer/set/upVector 0 1 0
#/vis/viewer/set/viewpointThetaPhi 50 165 deg
#/vis/viewer/zoom 1.6
#/vis/ogl/set/printMode pixmap
#/vis/ogl/printEPS

#/vis/open RayTracer
#/vis/viewer/set/upVector 0 0 1
#/vis/rayTracer/eyePosition 70 0 70
#/vis/rayTracer/trace


/vis/scene/add/hits
/vis/scene/add/trajectories
/vis/scene/endOfEventAction accumulate
```

Using DAWN - 2

- If you added the dawn executable to your PATH, it runs automatically
- Otherwise, cd to the dawn directory:

  ```
  ./dawn /path/to/WCSim/file.prim
  ```

- This link gives a good explanation of what the various GUI buttons do
- Produces PostScript file showing the detector
- Just a still image – have to regenerate if you want a different view
- If there are lots of PMTs, or a big detector, g4_00.prim can get big – 52m high, 20inch PMT, 40% photo-coverage nuPRISM is about 1Gb
- Can take a **long** time to load image in gv!
- But does show exactly what is in WCSim


```
O.x        =  0
O.y        =  0
O.z        =  0
camera_d   =  8784380
polar      =  0
azimuthal= 0
focal_d    =  259.057515
e_2d       =  1e-07
e_3d       =  0.001
```
RED:X, GREEN:Y, BLUE:Z (No hidden-line removal for the axes)

convertNuPRISM.C

- Back in WCSim
- 'root -l convertNuPRISM.C'
- 'convertNuPrism("inputfile.root","output.root", convert_fitqun ,copy_input)

- convert_fitqun – boolean to tell program whether to touch fiTQun tree
- copy_input – boolean to tell program to copy the input trees to the output file

- **UNTESTED!**

- Output file contains 'wcsimEx' tree, which holds the particle truth information in a viewable form

νPRISM

Work to be done

TRIUMF

- Need to validate convertNuPRISM.C and ensure it works with fiTQun output
- Check we have the geometry setup we want in WCSim
  - Verify positioning of detector components
- Process the NEUT files – these will be made available on the TRIUMF server shortly, and on the grid
  - Perform analysis studies!
  - Check detector simulation is correctly off-axis
- Make some particle gun files
  - Perform reconstruction studies!
  - Start optimising detector design