FairRoot & water cherenkov sim: Getting physics out of EsbRoot

- FairRoot and EsbRoot
- EsbRoot: what was done
- EsbRoot: examples
- Goals and wishes for this workshop

- All the work discussed here was done in close collaboration with Budimir Kliček
  - We also discussed the goals/wishlist and at least partially agrees

FairRoot: The underlying framework

- FairRoot is the software framework developed for the FAIR experiments/facility at GSI
  - Adopted by ALICE (LHC) for new Run3+ software
- FairRoot implements base functionality for large scale experiments: simulation, reconstruction, visualization and much more
- Useful reference: https://indico.in2p3.fr/event/17355/contributions/66940/

Why build on top of FairRoot

- We are a small collaboration so the more we can take over for free the better
- FairRoot is actively being developed and will be for 10+ years
- It seems to deliver two important things
  - Easy (but long) installation
  - Easy to set up (water cherenkov) simulation

EsbRoot Installation

- https://github.com/ESSnuSB/EsbRoot
  - See README.md for instructions
- 3 steps/packages to be installed
  - FairSoft
  - FairRoot
  - EsbRoot

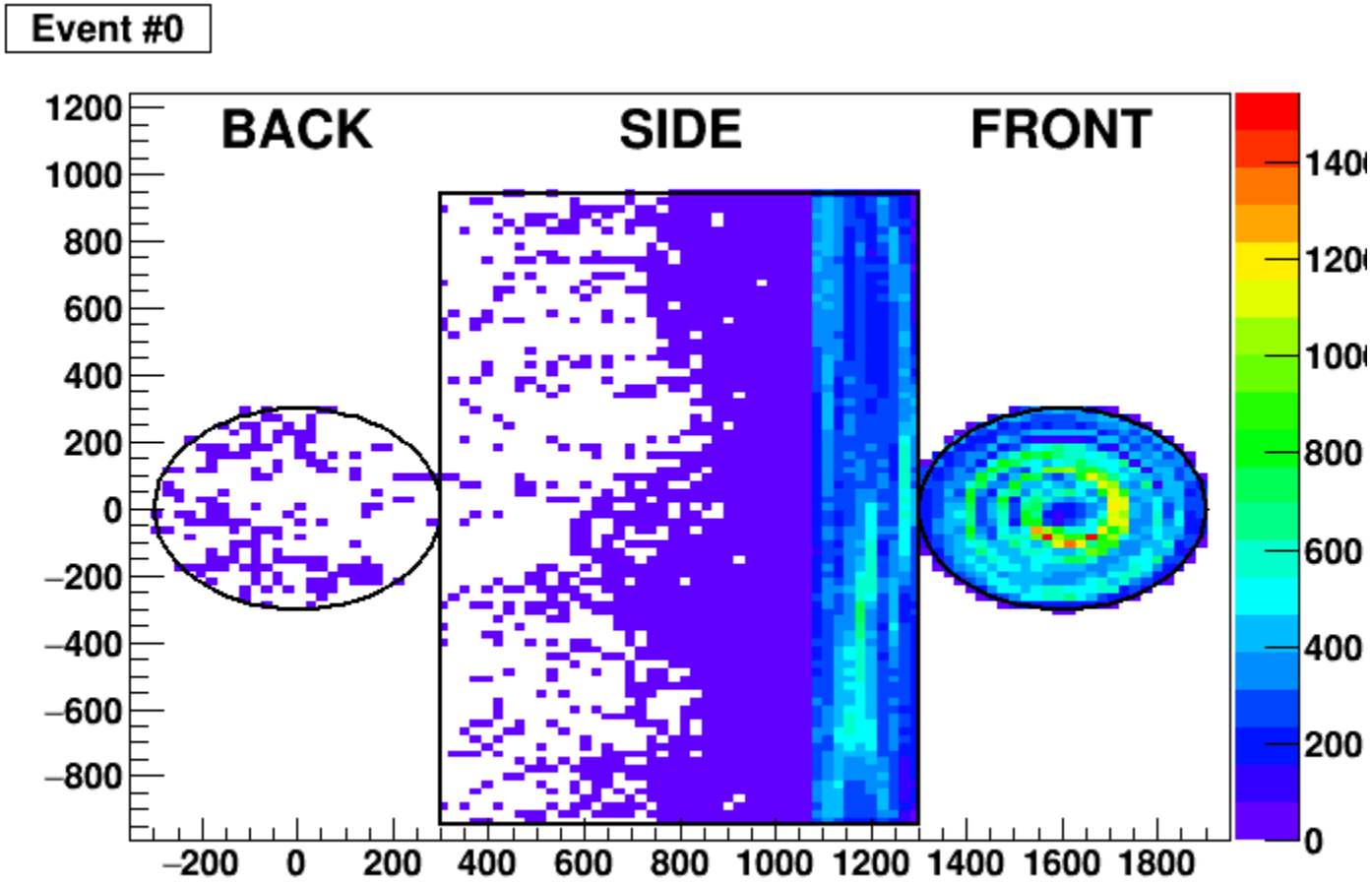
What was done

- Fast quick implementation (work in progress)
- Water Cherenkov volume
  - EsbGeometry/EsbWCDetector.cxx
- Photon hits
  - EsbData/EsbWCDetectorPoint.cxx
- Warning: not all classes in the generic FairRoot detector setup was implemented (could cause issues when more detectors are included)

EsbMacro/ess_sim.C

```
using namespace esbroot;

FairRunSim* fRun = new FairRunSim();
fRun->SetStoreTraj();
fRun->SetName("TGeant4"); // TGeant3/4
fRun->SetMaterials("media.geo");

FairModule *Cave= new EsbCave("CAVE");
Cave->SetGeometryFileName("cave.geo");
fRun->AddModule(Cave);
FairDetector *nearWc = new EsbWCDetector("NearWcDetector", kTRUE);
fRun→AddModule(nearWc);

FairPrimaryGenerator* primGen = new FairPrimaryGenerator();
fRun->SetGenerator(primGen);
FairParticleGenerator* partGen = new FairParticleGenerator(13, 1, 0, 0, 0.4, 0, 0, 150);
primGen->AddGenerator(partGen);

fRun->SetOutputFile(outFileName.Data()); // set output file
fRun->Init();

FairRuntimeDb *rtdb = fRun->GetRuntimeDb();
Bool_t kParameterMerged = kTRUE;
FairParRootFileIo* output = new FairParRootFileIo(kParameterMerged);
output->open("params.root");
rtdb->setOutput(output);
rtdb->saveOutput();

fRun->Run(nEvents);

fRun->CreateGeometryFile("geo_full.root");  // for additional full geometry file
```

EsbMacro/display_event_ND.C



More advanced FairRoot 3D event display that is not fully working can be found in:
EsbMacro/eventDisplay.C

Goals for workshop for ND and FD water Cherenkov (1/2)

- People should after the workshop be able to go back home and simulate a semi-realistic detector and start to work on the reconstruction
- This code will of course be further developed the next year. This means that things can change later

Goals for workshop for ND and FD water Cherenkov (2/2)

- Provide a set of quite generic water cherenkov classes (detector, hits, digitizers – one of them to be nicely coded and documented as an example) and a flexible generator input (FairRoot)
- Simulation should be able to produce
  - Track information
  - Photon information (GEANT hits)
  - PMT information (digitization)
    - But not necessarily realistic GEANT PMTs.

Wishlist: need to work on (1/4) Generator

- FairRoot provides simple generators
- We need a generator that can give us neutrino events
  - We provide neutrino specie, momentum and interaction point, we get realistic electron/muon momentum
  - Should be based on GEANIE

Wishlist: need to work on (2/4)

- GEANT4 parameters for water
  - Need realistic values and validation (media.geo). WCSim and MEMPHYS.
- Hit information
  - Do we store the right information
  - Some restrictions at it has to derive from FairRoot base class
- Track information
  - Do we store the right information

Wishlist: need to work on (3/4) Digitization

- Ideally define it as a FairRoot module that can be run separately (post processing) or as part of the simulation
- I suggest a very simple implementation where one sums up hits in a circle on the surface. Still one needs to define the layout and geometry in a smart way (have a look at MEMPHYS and WCSim)
- Need a semi-realistic PMT response

Wishlist: need to work on (4/4) Reconstruction/Framework

- A simple reconstruction example
- So that one can understand how to access data, geometry and so on

- Implement another detector and see that they can work together

Wishlist: nice to work on

- Comparison between simulation results from EsbRoot and MEMPHYS or WCSim
- Advanced reconstruction
  - Would be nice also to check that simulation outputs the needed information
- Get the 3D visualization to work correctly
  - Likely only possible for PMTsignals as there are too many

Backup

How to charge GEANT4 compile flags in FairSoft

- If one wants to use FairSoft for other software packages, e.g., WCSim it can be useul to know that one can edit the file: FairSoft/scripts/install_geant4.sh before doing the installation with configure.sh
- We will likely solve this by making our own copy of FairSoft and FairRoot