Helpnote 2 - Run the QPIX Software on HEP

Till Dieminger

We use the following terms:

- Server Side: The Manchester HEP cluster at `<hostname>`
- Client Side: Your local machine

GEANT 4 Part

This part is responsible for generating the interactions of the particles in the detector volume and to produce the subsequent particles - like dirft electrons.

Clone

On the server side, navigate to your source folder (best situated in the data dictionary of your cluster machine), run

```
git clone https://github.com/Q-Pix/qpixg4
```

Enter the `qpixg4` directory and run

```
source setup/setup_cvmfs.sh
```

To `qpixg4/CMakeList.txt` you have to add two lines, which allow the MC generator `MARLEY` to be linked. For this add

```
include_directories($ENV{MARLEY_INC})
link_directories($ENV{MARLEY_LIB})
```

to the CMakeList.txt file, such that the end of the file reads

```

## include ROOT header files
include(${ROOT_USE_FILE})

## link ROOT libraries
link_libraries(${ROOT_LIBRARIES})

## MARLEY stuff
include_directories($ENV{MARLEY_INC})
link_directories($ENV{MARLEY_LIB})
```

```
## Recurse through sub-directories
add_subdirectory(src)
add_subdirectory(app)
add_subdirectory(cfg)
```

Now navigate to the `qpixg4/Build` directory and run

```
cmake ../
```

and after that

```
make
```

If all of this runs without any issues, hurray we are done. If not, contact some other person who did this already and ask them to update this file!

Usage

As every Geant4 simulation, you run it by the build version. Navigate to `qpixg4` and run

```
./Build/app/G4_QPIX macros/<your-macro>
```

For the macros, you can edit it starting from the examples in the `qpixg4/macros` folder.

RTD

This part uses the output of the G4 simulation and drifts the electrons through the detector material and adds the Q-PIX readout.