gSeaGen Style: First Look

J. Coelho

15 March 2024

Enforcing Style

- The easiest way for us to enforce a style is to use automated tools
- One tool is clang-format, which has several C++ style options
- Swim is currently using a custom style from clang-format
- At first glance, the gSeaGen code currently is closest to LLVM

```
Swim:
 60 files changed, 12930 insertions(+), 10952 deletions(-)
LLVM:
 60 files changed, 12244 insertions(+), 10770 deletions(-)
Google:
 60 files changed, 12210 insertions(+), 11136 deletions(-)
Mozilla:
 60 files changed, 13682 insertions(+), 10580 deletions(-)
Chromium:
 60 files changed, 12678 insertions(+), 11026 deletions(-)
WebKit:
 60 files changed, 14458 insertions(+), 14456 deletions(-)
Microsoft:
 60 files changed, 16467 insertions(+), 14494 deletions(-)
GNU:
 60 files changed, 17877 insertions(+), 13534 deletions(-)
```

Enforcing Style

- Mostly changing spacing and indentation in a consistent format

```
namespace genie {

class EvtWrite {

public :

  EvtWrite(GenParam * GenPar, bool OptPDG);
  EvtWrite(GenParam * GenPar, bool OptPDG, event * evt);

  ~EvtWrite();

  void                  WriteHeader(void);
  void                  WriteEvent(GSeaEvent * SeaEvt);


protected:


  event *               fEvt;
  ofstream              fOutFile;

  bool                  fNewEvt;

  GenParam *            fGenPar;

  TDatabasePDG *        fPdg;

  bool                  fOptPDG;


};
} // genie namespace
```

```
namespace genie {

  class EvtWrite {
    public:
      EvtWrite(GenParam* GenPar, bool OptPDG);
      EvtWrite(GenParam* GenPar, bool OptPDG, event* evt);

      ~EvtWrite();

      void WriteHeader(void);
      void WriteEvent(GSeaEvent* SeaEvt);

    protected:
      event*   fEvt;
      ofstream fOutFile;

      bool fNewEvt;

      GenParam* fGenPar;

      TDatabasePDG* fPdg;

      bool fOptPDG;
  };
} // namespace genie
```

Checking for C++ Issues

- Another useful tool is cppcheck, which searches for non-conformant code that could lead to bugs
- This is stuff like uninitialized variables, possible memory leaks, etc.
- Found a few issues that should be easy to fix

```
src/OutputWriters/KM3NeTWrite.cxx:290:28: error: Undefined behavior:
Variable 'ch' is used as parameter and destination in sprintf(). [spr
intfOverlappingData]
      sprintf(ch, "%s %s", ch,
                           ^
src/PropaMuon/PropaMuon.cxx:605:10: error: Uninitialized variables: t
rackf.Id, trackf.MotherId, trackf.Pdg, trackf.Status, trackf.Length,
trackf.E, trackf.T [uninitvar]
  return trackf;
         ^
src/PropaMuon/PropaMuon.cxx:605:10: error: Uninitialized struct membe
r: trackf.Id [uninitStructMember]
  return trackf;
         ^
```

General Comments

- I'm not that much of an expert in C++, but here are some issues that could be addressed in my opinion
  - Use spaces, not tabs (automatic with clang-format)
  - Cleanup unnecessary includes
  - Avoid includes in headers if they really apply to source code
  - Avoid using namespace in headers
  - Avoid pointer members if your class owns this object (easier cleanup)
  - Cleanup commented out code and non-documentation comments

```
#ifndef _EVTWRITE_H__
#define _EVTWRITE_H__

#ifdef _ANTARES_ENABLED__

#include <cstdio>
#include <ctime>
#include <fstream>
#include <map>
#include <string>
```

```
using namespace std;
using namespace genie;

namespace genie {

  class EvtWrite {
    public:
      EvtWrite(GenParam* GenPar, bool OptPDG);
      EvtWrite(GenParam* GenPar, bool OptPDG, event* evt);

      ~EvtWrite();

      void WriteHeader(void);
      void WriteEvent(GSeaEvent* SeaEvt);

    protected:
      event*   fEvt;
      ofstream fOutFile;

      bool fNewEvt;

      GenParam* fGenPar;

      TDatabasePDG* fPdg;

      bool fOptPDG;
  };
} // namespace genie
```

```
if (fGenPar->TrackEvts > 0) {
  sprintf(ch, "gSeaGen %s %s", fGenPar->PropCode.c_str(),
          fGenPar->PropCodeVer.c_str());
  fEvt->taga("physics", ch);
}

//  sprintf(ch, "%f", fGenPar->TGen);
//  fEvt->taga("tgen",ch);

if (fGenPar->GenMode.compare("BIN") == 0) {
  sprintf(ch, "%10d", fGenPar->Primary);
  fEvt->taga("primary", ch);
}
```

Backup