Unittesting and Coin 3

- Why Boost.Test?
  - Wanted integrated/inlined framework to avoid extra configure tests, link dependencies.
    - Boost.Test, TUT
  - Parts of Boost already on its way into Coin 3.
  - TUT seem too simple and inflexible.
  - Safer to bet on Boost being maintained and developed than the lesser known TUT.

Unittesting and Coin 3

- How?
  - Custom setup/architecture you will probably only find in Coin.
  - Test code extracted from between #ifdef-tag-wrappers inside the Coin source code and compiled separately (testcode in Coin sources is not compiled into Coin).
  - The makefile decides which tests are included/run.
  - Makefile updates must be triggered manually, at which point user-specified filtering can be applied.

Unittesting and Coin 3

- Advantages
  - Very convenient test-code locality.
  - Forces header file order convention.
  - Tests run on uninstalled binary.
  - Testsuite can be stripped down to just the tests of interest.
- Disadvantages
  - Only public API is testable.
  - Adding tests to virgin files must be followed by a makefile update.
  - Loose tests goes where? (whereever)

Example

```
[From the bottom of Coin/src/geo/SoGeoSeparator.cpp]

#ifdef COIN_TEST_SUITE

BOOST_AUTO_TEST_CASE(initialized)
{
  BOOST_CHECK_MESSAGE(SoGeoSeparator::getClassTypeId() != SoType::badType(),
                      "SoGeoSeparator class not initialized");
  boost::intrusive_ptr<SoGeoSeparator> node(new SoGeoSeparator);
  BOOST_CHECK_MESSAGE(node->getTypeId() != SoType::badType(),
                      "SoGeoSeparator object wrongly initialized");
}

#endif // COIN_TEST_SUITE
```

- Note exact #ifdef-wrapper that the extracting script looks for.
- All test-cases use BOOST_AUTO_TEST_SUITE(token) for function wrapper.
- BOOST_CHECK_MESSAGE(), BOOST_CHECK_EQUAL(), BOOST_CHECK(), BOOST_REQUIRE(), BOOST_ERROR(), BOOST_WARN_MESSAGE(), and lots more are defined in Coin/include/boost/test/test_tools.hpp.
- Multiple test-cases and #ifdef blocks can be randomly placed in the source files.

Running tests

```
Coin$ cd testsuite
testsuite$ make
[extraction, compilation, linking...]
LD_LIBRARY_PATH=../src/.libs:$LD_LIBRARY_PATH \
        DYLD_LIBRARY_PATH=../src/.libs:$DYLD_LIBRARY_PATH \
        PATH=../src:$PATH \
        ./testsuite --log_level=warning --show_progress=yes \
          --detect_memory_leaks=0

0%   10   20   30   40   50   60   70   80   90   100%
|----|----|----|----|----|----|----|----|----|----|
Running ### test cases...
***************************************************
Coin$

Coin$ make verbose

Coin$ make makefile-update filter=SoGeo
```

Unittesting and Coin 3

- Current status
  - Framework operational and proven to work.
  - Some support/utility functions are in place.
    - Room for lots more.
  - Buildbot running the testsuite when build succeeds.
  - No real useful testing being done, no test-driven development or bugfixing happening yet.
  - Coin 2 is Boostless and consequently testsuiteless (and will remain so).