Building, testing and sharing a C++ library

Luiz Irber
@luizirber
Why?

- Play with C++11
- Explore unit testing solutions
- How do I distribute my code?
Minifloat

- IEEE 754 (floating point spec)
- 8 bits
  - 1 sign bit
  - 4 exponent bits
  - 3 mantissa bits

"In place of infinity, we usually put some really big number, like 15."

Anonymous Computer Science Professor
Why is a minifloat useful?

- khmer Count-Min Sketch
  - Probabilistic data structure
  - currently limited to 0-255 (8 bits/unsigned char)
- Input data
  - Overrepresented k-mers
- Minifloat might be a solution?

"In place of infinity, we usually put some really big number, like 15."

Anonymous Computer Science Professor
First implementation

- Lots of bit shifting
- Conversion from int/float to minifloat
- Single header, easy to distribute

How do I check if it is correct?
- Only 256 possible combinations
- Exhaustive testing is fast enough
Unit testing: Catch
https://github.com/philsquared/Catch

- Single header
- JUnit output
- No external dependencies

```cpp
#define CATCH_CONFIG_MAIN // This tells Catch to provide a main() - only do this in one cpp file
#include "catch.hpp"

unsigned int Factorial( unsigned int number ) {
    return number > 1 ? Factorial(number-1)*number : 1;
}

TEST_CASE( "Factorials are computed", "[factorial]" ) {
    REQUIRE( Factorial(1) == 1 );
    REQUIRE( Factorial(2) == 2 );
    REQUIRE( Factorial(3) == 6 );
    REQUIRE( Factorial(10) == 3628800 );
}
```

https://github.com/philsquared/Catch/blob/master/docs/tutorial.md
Continuous Integration: Travis

- [https://travis-ci.org](https://travis-ci.org)
  - Free for open source
  - Private repos: free with the GitHub Student Pack
- Easy to configure
  - `.travis.yaml`
- Integration with GitHub
  - Can set PR and commit status
- Integrated deployment
  - Heroku
  - biicode
Travis issues...

- **Outdated compiler versions**
  - GCC 4.6 (!?!?!??!?!?!)
  - clang 3.4

- **Solution: PPA**
  - [http://stackoverflow.com/a/30925448/2631881](http://stackoverflow.com/a/30925448/2631881)
  - [http://llvm.org/apt/](http://llvm.org/apt/)
Dependency management

● **Ecosystems**
  ○ Python: PyPI
  ○ Ruby: Rubygems
  ○ Node: NPM
  ○ Rust: Cargo

● **Why not C/C++?**
  ○ Ad-hoc solutions
    ■ Autoconf
    ■ CMake
    ■ Cargo cult programming…
  ○ CMake offers good infrastructure
Dependency management: biicode

- biicode.com
- C/C++, Raspberry Pi, Arduino, Node.js
  - Emscripten on the way
- Spanish startup
  - Academia experience
    - working with robotics
    - Professor, left tenure position
  - Closed source
Dependency management: biicode

- Open source (May 2015)
- Based on CMake
  - Leverage CMake infrastructure
  - Add declarative configuration
    - (still CMake, you can customize)
- Nice docs
  - [http://docs.biicode.com/c++.html](http://docs.biicode.com/c++.html)
  - [http://docs.travis-ci.com/user/deployment/biicode/](http://docs.travis-ci.com/user/deployment/biicode/)
  - [http://blog.florianwolters.de/educational/2015/01/11/Adding_biicode_support_for_the_library_Catch/](http://blog.florianwolters.de/educational/2015/01/11/Adding_biicode_support_for_the_library_Catch/)
Biicode issues

● Directory structure is a bit confusing
  ○ I would like to have blocks and code in the same repo
  ○ using ‘bii’ command helps

● Needs more users
  ○ https://www.biicode.com/florianwolters/catch or https://www.biicode.com/Manuzor/catch?
  ○ Ideally library authors should maintain the block
  ○ Homebrew or GitHub?
Getting started with biicode

http://docs.biicode.com/c++/gettingstarted.html
Can we automate everything?

http://www.commitstrip.com/en/2015/06/22/can-we-automate-everything/
Making minifloat useful: Let’s support arithmetic!

- Hmm...
- Standard: 70 pages long
- Operations: 13 pages
- (Is there a shortcut?)
OpenEXR to the rescue

- half float (16 bits)
- very clever implementation
  - cache unions -> fast conversion
  - use native float for operations
- Nice comments
  - doubling as documentation
- BSD License
- (but I’m still grokking it)

https://github.com/openexr/openexr/blob/master/IlmBase/Half/half.h
Thanks!

code: github.com/luizirber/minifloat

This talk: https://goo.gl/Fp0I4G

@luizirber
Cunningham's Law

"the best way to get the right answer on the Internet is not to ask a question, it's to post the wrong answer."

https://en.wikipedia.org/wiki/Ward_Cunningham#Ideas_and_inventions