Code Review: Stars

John Jannotti

CSCI032 (cs0320)

Feb 15, 2017
Course Announcements

- Autocorrect is due in 12 days.
  - Keep a steady pace
- Javascript lab is going on now.
  - Trie is probably simpler than KdTree.
  - But the Javascript part is new and strange.
- No class next Tuesday,
  - Please don’t take off from now until next Wednesday.
Everyone’s code is horrible

The only valid measurement of code quality: WTFs/minute

(c) 2008 Focus Shift/OSNews/Thom Holwerda - http://www.osnews.com/comics
Except for my guinea pigs

- Thanks to a few students for letting me use their well-done assignments as examples.
- They’ve got a lot of nice stuff in them.
- We’ll still find ways to improve them.
  - That’s always true.
- I had more volunteers than I can cover. I’ll put you at the from of the line for Autocorrect (if you still want it!)
“Maslow’s code review”

http://blog.d3in.org/post/111338685456/maslows-pyramid-of-code-review
Focus on what matters

- Look for
  - potential bugs.
  - inaccuracies (comments, specs).
  - inconsistency (internally, or with project)
  - unneeded reimplementations.
  - possible simplifications

- Try to avoid “bikeshedding”
  - Sometimes there are two ways to do things.
  - Try to focus on differences that matter.
  - and project consistency.

- Care most about things that are hard to change later.
  - You will have “users” who count on things not changing.
Nice things we looked for

- KDTree is generic
- KDTree implements Collection, extends AbstractCollection
- KDTree is n-dimensional
- Handle commands well (switch, data dispatch)
- Operates on squared distances
- Naïve implementation for testing
- Provide iterator() on KdTree, implement Iterable
- Leaf Nodes contain points that can be searched linearly
- Closes the file (+1 more for doing it right)
- Uses a Map to catalog star names for easy lookup
- Line numbers in error messages for bad star files
- Uses @Override for methods
- Uses assert/some other equivalent in their source code (not tests)
- Report multiple bad lines in one file
Project stats

- A — 29 files, 1483 lines (958 main / 525 test)
- B — 17 files, 1415 lines (1135/280)
- C — 21 files, 1452 lines (974/478)
- staff — 16 files, 1144 lines (909/235)

Generated by David A. Wheeler’s “SLOCCount”