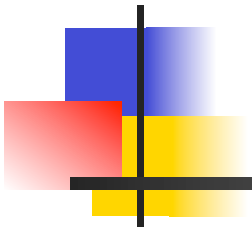
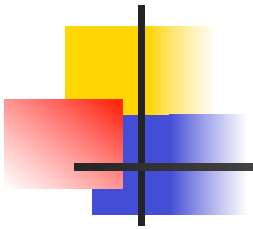


Pfff: PHP Program Analysis at Facebook

Yoann Padioleau (Facebook)

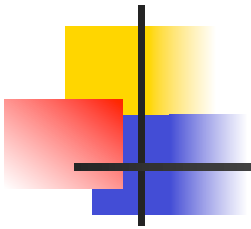
http://github.com/facebook/pfff

OCaml Users and Developers Workshop 2013



About this talk

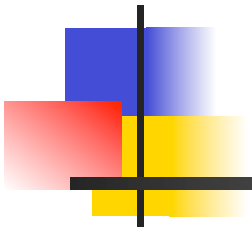
- Feel free to ask questions during the talk



PHP Program Analysis at Facebook

- Deadcode removal (global analysis)
- Test coverage (dynamic analysis)
- Use/Def checker (e.g. use of undefined function)
- Variable checker (e.g. use of undeclared variable)
- Syntactical Grep lint rules
- Tainting Analysis via abstract interpreter (XSS)
- Type checker daemon (Julien's talk at CUFP'13)
- Separation Logic? (Monoidics ocaml startup acquired)

But I will not talk at all about any of this in this talk

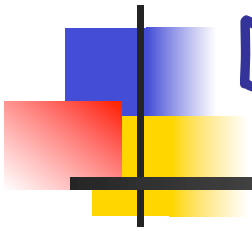


Pfff: Tools to Help Understand Large Codebase

Yoann Padioleau (Facebook)

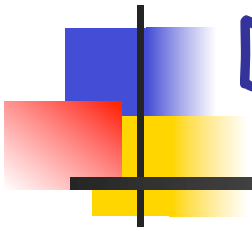
http://github.com/facebook/pfff

OCaml Users and Developers Workshop 2013



Demo Codemap

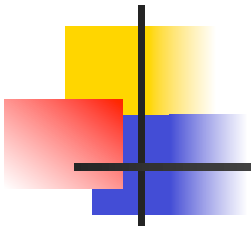
- Google maps for source code
- Program analysis + software visualization
- Need a 30' monitor to really appreciate



Demo Codegraph

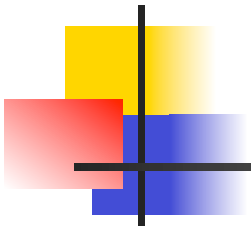
- Focus on code relationships, not source code
- Understand the "Software Architecture"

Demo Codemap + Codegraph



Pfff tools and APIs

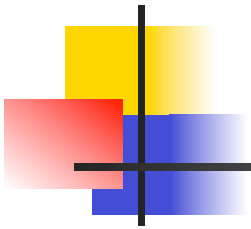
- Other tools in Pfff:
  - CodeQuery: Prolog queries on codebase
  - stags: precise TAGS generator
  - sgrep/spatch: syntactical grep and patch
  - scheck: linter
- Program analysis APIs for many languages (parsers, AST visitor/dumper/matcher/highlighter, use/def global analysis, coverage analysis, refactoring, etc):
  - OCaml (thx to .cmt in 4.00)
  - PHP
  - Java (thx to joust and javalib)
  - C/C++ (thx to yacfe and clang)
  - Html, Css, Javascript
  - ...



Conclusion

- CodeMap: a scalable semantic-based source code visualizer/searcher/ navigator
- CodeGraph: a scalable dependencies visualizer
- Future work:
  - Reordering to minimize backward deps
  - Web UI (most of it done, with ocsigen)

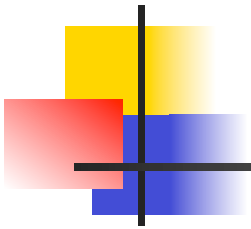
$ git clone git://github.com/facebook/pfff.git



Features

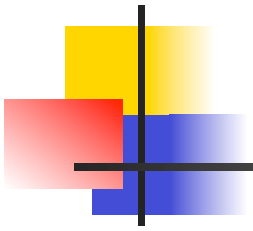
- Big picture, treemaps, "macro level"
- Search, navigation
- Semantic-based code highlighting "micro level"
- Connection to editor (emacs/vim)
- Extensible via layers (predator mode)

Codemap is not an editor



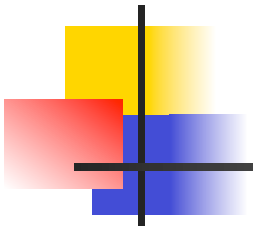
Features: treemap

- Each rectangle is a file
- Size of rectangle =~ size of file
- Color of rectangle = "aspect" (test, main, storage, security, etc)
- "Code aware" (heuristics)
  - Auto generated file do not eat real-estate
  - Code is more important than data, xml, jpg
- Tiling, use all the space



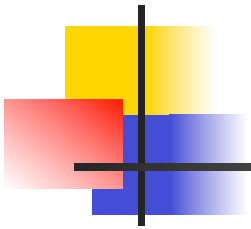
Features: search and navigation

- Search
  - Highlighted rectangle
  - Ranked entities
- Navigation
  - Up/down (not as smooth as google maps)
  - Direct access to file (faster than speedbar or expand-directory widgets)
  - Can see spread directories



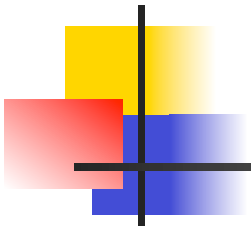
Features: semantic code visualizer

- grammar-based highlighting, not regexps as in emacs/vim
  - Know records vs functions vs constants
  - Functions/classes are in bigger size than statements
- Tiling, use all the space, multi columns
- Semantic aware (global analysis)
  - Important functions are in bigger size



Layers: alternate color schemes

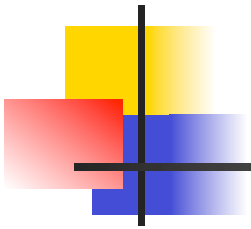
- Age (help find dead code)
- #authors (important stuff usually)
- Activity (what's going on?)
- Code coverage
- Bugs/warnings of linter
- grep/sgrep results
- Top/Bottom modules
- …



Conclusion

- A semantic-based source code visualizer/searcher/navigator
- Accelerate loading the code into your brain (can see 20 files at once)
- Future work:
  - Smoother zoom

$ git clone git://github.com/facebook/pfff.git



Related work

- SeeSoft (does not scale, no treemaps)
- Code Thumbnails (2 different modes)
- 3d visualization (not sure it helps, eat pixels)
- Disk explorer (not code aware, no micro-level)