GIST A6

BY ANDREW SIKOWITZ
OVERVIEW FOR A6

• The first sprint, out of three, of the midterm project!
• Come up with a vision of what you will build
• Build functionality that can be demoed
• Demo it to your TAs
• Write a progress report reflecting upon the sprint, and preparing for the next one
• Meant to mimic software development in the workplace
  • You are not (necessarily) given a formal specification of what to build, nor formal evaluation metrics
A6 DELIVERABLES

- New Team Expectations Agreement
- Source code in a zip file, due Wednesday (one late day allowed)
  - Authors compilation unit, with <authors.ml> and <authors.mli> as before
  - No _build directory
  - Makefile please
- Demo (in discussion section, Wednesday or Thursday)
- Progress report, due Thursday (no late days allowed)
Project Requirements

- **Built something cohesive**
  - An single application or system that does something; a collaborative effort of all team members

- **Build something substantial**
  - Expect an end product equivalent or greater than that of A2+A3 or A4+A5

- **Built it from scratch**
  - Don’t build off work from other classes
  - Code from previous assignments is not counted towards what you have built
  - Use only the libraries used in A0-A5 (listed on website)
    - This means no networking, GUIs, multithreading, Jane Street libraries (e.g. Async)
    - This may seem lame, but will really help your workload
Example Projects

- Turn-based game with text interfaces
  - There are a LOT of board games (and non-board games) to choose from
  - ANSITerminal, which you can use, allows for pretty nice text interfaces

- Computational engine
  - For example, one team built an application that performs a variety of statistical analyses

- Anything you want!
  - Just make sure it's in the middle ground between trivial and impossible
Sprints

- The “agile” method of software development, as opposed to “waterfall”
  - Adaptive planning, evolutionary development, early development, continual improvement
- Forces you to iteratively build something that works
  - Fewer “we couldn’t piece it together at the end” issues
- Receive feedback weekly; set goals weekly
- Allows you to change what you are building in a more structured way
- You will have 3 total, producing successive weeks: an alpha, a beta, and a finished product
Progress Reports / Grading

• Evaluate your group’s progress in the past sprint, each week
  • Outline what you did, what has changed, and what you plan to do in the next sprint
  • Grade your group’s delivered work for that sprint, and coding standards
  • Single progress report for a group

• Your section TAs will evaluate your sprint based on the demo and progress report

• Very coarsely graded: Satisfactory, Good, or Excellent
  • Stay on top of things; don’t bite off more than you can chew. If you do that, you’ll be fine.
Final Tips

• You are given a lot of flexibility with the midterm project
• You’re also working for the first time on a 3-week project
  • Maintain your code: it’s easier to write clean code once than write messy code and fix it later
  • Try not to fall behind
    • If you submit late for the first assignment, it’s a lot easier to submit late for the rest too
  • Plan ahead… but the sprint process should make things flexible
• Have fun! It’s an opportunity for you to build something that you want that’s cool