Recitation #12

Friday 20th-Nov-2015

Electrical & Computer ENGINEERING

Carnegie Mellon

Note: Course slides shamelessly stolen from lecture

Announcements and Administrative Stuff

- **Exam #2 on Wednesday 2nd December**
- **Project 12 due Monday 7th December**
- **Project 13 due Tuesday 15th December**
  - Demos due by the 5 PM on Monday the 14th

E-mail Check-list (On Admin Page)

- **Before writing that e-mail**

  Check blackboard to see if an answer has been posted

  Re-read the assignment to make sure you are reading it correctly

  Look at the grading checklist to see if it has relevant information

  Look at the Pepsi machine example to see if it provides a reasonable example

  Discuss the problem with your teammates and see if you can agree upon a reasonable way to proceed without violating written assignment requirements

- **Regarding e-mail on assignments**

  If you simply don't understand, then skip the e-mail and go to office hours

  If you think there is a defect in the course materials, include the URL of the document you have a question about and a specific explanation of the defect or contradiction

  Start your e-mail with "I've used the e-mail question checklist, and I think the following is an issue:" or the e-mail might not be replied to

  Wait 5 minutes before sending. Seriously. We get lots of "oops, found it" e-mails less than 5 minutes after sending a query

Proj 11: RT-9

- **“What does this even mean?”**
  - That’s a very good question!
- **Customer requirements are often like this.**
- **Possibilities from “maximum practicable”**
  - Commit point calculation?
  - How long at slow speed?
  - Hoistway position windows for speed transitions
  - So many others…
- **Our grading monitor is fairly loose**
  - All of the described methods will ensure you pass our monitor

Project 12

- **Test the heck out of stuff!**
- **All unit tests must pass**
- **All integration tests must pass**
- **Run acceptance tests**
  - Write an acceptance test generator
  - All acceptance tests **should** pass
  - **Minor Update:**
    - **Any acceptance test that does not pass must have an explanation in the notes section, including Issue Log Entry number**
  - Use -b 200 and -fs 5.0
- **Validate your monitors**
- **Update traceability**

Test Generation

- **We provide a bash template…**
  - … that you are welcome to ignore
- **However, we must be able to run it on the ece0xx machines**
  - If it's compiled code, give us source and:
    - A makefile to build it, or
    - code compiled on the ece0xx machines
  - If it's a script make sure it's in a language the ece0xx machines support
- **Must create "reasonable tests"**
  - Tests **should** be similar to the project 8 test
  - Randomness will mean that not all of them will be but
    - If we run it 10 times and no test meets the requirements, it's no good.

Course Project Exit Criteria

- **Must have a working elevator to complete the course**
  - Run Time Monitor Must Be Implemented
    - Pass all unit tests with zero failed assertions
    - Pass all integration tests with zero failed assertions
  - Pass all acceptance tests
    - Using -b 200 and -fs 5.0
    - Zero failed assertions (after startup)
  - Pass The Demo for TAs
- **Non-working results in Incomplete if you don't get it working by grade deadline**
- **+1% final grade for best elevator (one group only)**
  - Rank groups by average performance and satisfaction across acceptance tests
- **+2% final grade for complete and consistent design portfolio**
  - All groups are eligible for this

Suggestions for Project 13

- **Impose a "code freeze" as soon as possible**
  - Stop changing code as soon as your design reliably passes all tests
  - No new features, no new fixes, no new comments or cleaning
  - If it works, stop fixing it, and archive a COMPLETE COPY somewhere safe
    - "Safe" does not mean in the same directory structure as your 649 working copies
    - "Safe" means write-protected
    - Version control is very effective, but nothing is perfect!
    - Keep all your different "this one works" copies until you get your final course grade
  - If you edit a single line of code (even a comment), re-run ALL tests

- **When submitting your code**
  - Do a clean export from version control and upload it
  - Extract a new clean copy of the simulator framework in a new directory
  - Download what you submitted and insert it into the new simulator
  - Recompile (make clean) and re-run all your tests

- **Focus on traceability as soon as you can (its worth A LOT of points)**
  - End to End traceability is required for this project
  - Tweak performance after you have a clean, traceable hand-in so you don't get caught short on time

Questions?