

CS473 Practicum in Artificial Intelligence

Fall 2000

Status Reports and Code Reviews

General Suggestions for Completing the Project. To encourage that you write your code in stages rather than waiting until the end of the semester, you'll have to turn in a status report and copies of your code for review twice during the semester — whether or not it's running. A portion of your final grade is based on these intermediate project reviews. The format of the status reports is included below, but I'll preface that with some general suggestions for how to go about constructing your CS473 systems.

- First, think about the major “modules” that your project requires and the knowledge/data structures (e.g., rules, state descriptions, board representations, graphs, networks) that you'll need. If at all possible, implement the modules as “stubs,” i.e., functions that have no substantive code, but that specify all of the necessary parameters and that make calls to the appropriate module stubs. (Ideally, these should be completed for code review 1.)
- Write the code to implement your knowledge/data structures. This code normally takes the form of a set of abstract data types with all of the appropriate constructor and accessor functions. (Again, ideally these should be completed for code review 1.)
- Get the system running from beginning to end as soon as possible — just on a simple example. This will invariably require making many simplifying assumptions for major chunks of the system. (It should be possible to complete this for code review 2.)
- Design and implement any inference engine or control strategies needed for your system, e.g., specific search strategies, a rule interpreter, move generator, learning algorithm. (Completed by project end.)
- Incrementally remove the simplifying assumptions by adding the necessary code. Evaluate the system as set forth in your project proposal. (By project end.)

What to turn in for the status reports.

Due Dates: Wednesday October 25 and Monday November 13 at the beginning of CS472 class time. If you have your report ready early, you can give it to me in CS472 or drop it off at my office during the day or leave it in my mailbox (4th floor Upson near the main office) or give it to any of the TA's.

Format: Only one status report per project should be turned in. It does not need to be long (1 or maybe 2 pages should suffice). It should contain the following sections:

1. A short (one or two sentence) **summary of the project goal/topic**. (This is just to remind me what you're working on.)
2. A list of **the major components of the system** as well as **the current status of each** (e.g., designed, implemented, tested, finished, almost finished with coding). For the pieces not yet finished, indicate tentative completion dates. The schedule doesn't have to be incredibly detailed, but it should include entries for all substantial modules/sub-modules of your system.

3. A list of **the major steps in your planned evaluation** of the system and **the status of each**. For some projects it may be easier to merge this list with the component/module status list above.
4. Optional: an informal log your efforts. If you've been keeping notes on your progress, feel free to turn these in as well. These might include: (1) notes on what aspect of the project you worked on and when, (2) what was accomplished in each "session", (3) design questions, (4) implementation questions, (5) descriptions of experiments and results, (6) problems that arose, (7) anything else that you think is relevant to the project.
5. Also e-mail to me (cardie@cs.cornell.edu) a copy of the code that comprises your project. Send it as a zipped or gzipped file. If you are using existing code libraries or previously written code, be sure to make clear which parts of the code are new for the CS473 project. (I really only need to see the code that you're writing for the project, NOT any of the code that you're making use of but were not written as part of the CS473 project.)

The grade for the status report will be based on: (1) progress made; (2) clarity of the report; (3) algorithm design; procedural and data abstraction; (4) readability, style, documentation of the code. If your final project grade is better than a status report grade, I will replace the lower status report grades with the final project grade.