Making Lemonade:
Exploring the Bright Side of Large Lecture Classes

Steven A. Wolfman
Computer Science Department
University of Washington
Seattle, WA 98195
wolf@cs.washington.edu

Abstract

Pedagogy of large lecture classes has traditionally focussed on deemphasizing the problems their size creates. This approach has yielded valuable practical advice for instructors. However, this paper argues that there are pedagogical advantages to the large lecture format and that exploiting these advantages can *further* improve classroom instruction. I present some advantages of large classes and anecdotes that demonstrate how to exploit these advantages.

1 Introduction

At a college football game in 1981 cheerleader Rob Weller encouraged audience members to stand up in a moving circular pattern that traveled around the stadium. The result was both visually arresting and exciting for the crowd, and it became known as "the Wave" [8].

The Wave would never have worked in a crowd of twenty. The atmosphere of the crowd and its sheer numbers enabled a massive form of interaction that is not compelling in a small group. Could the same hold true in large classrooms? Might there be pedagogically valuable exercises and methods that work well in large classes — and even better in larger classes?

Conventional wisdom in computer science holds that the answer is no. The prevailing opinion is that attributed to Phil Wankat: "anything you can do in a large class you can do better in a small one" [4]. Large lectures are seen as an ugly economic necessity born of resource restrictions. In response, the education community has produced reams of material on techniques for teaching large classes [1, 2, 3]. These materials are invaluable in their intended purpose: to soften the negative impact of large class size.

Yet, the same spirit that enlivens a football game *can* manifest itself in the CS classroom [7]. Unfortunately, the advantages of the large lecture format have received little attention in the education literature.

In this paper, I present an approach to the pedagogy of large lecture courses that exploits these advantages. First, I enumerate several aspects of large classes that offer pedagogical advantage. Then, I give a list of advice — based on anecdotes from a large CS course — on how to exploit these advantages in practice. I conclude with a challenge to the CS education community to energetically explore the positive pedagogical aspects of large lecture courses.

It is important to emphasize before going on, however, that the strategy described in this paper does not rule out the conventional strategy: avoiding the disadvantages of large classes. Rather, these strategies are complementary. Neither does this paper support large classes over smaller ones; such a comparison is irrelevant to the fact that many classes *are* taught — and will likely continue to be taught — as large lectures.

The anecdotes in this paper come from the Spring quarter offering of CSE142 at the University of Washington. The course was a 10 week introductory computer science course (i.e., CS1). Two lecture periods of the class were offered in parallel. I taught one while my colleague Professor Martin Dickey taught the other.[1] My class's size was two hundred five students. The enrollment included CS premajors and over two dozen other majors from Accounting to Zoology. Prior programming experience varied from none to extensive. The course met weekly for three one hour, full-class lectures and a one hour section in groups of approximately twenty students each. The students shared lab space staffed with consultants. The course had electronic communication

Published in Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education, 2002.

[1] I am indebted to Prof. Dickey as well as previous quarters' instructors for guidance and collaboration on many aspects of the course.

through an unmoderated newsgroup and staff mailing list as well as a moderated class mailing list. The class was well received by students with student evaluations in the top 10% of all engineering courses at U. Washington.

2 Advantages of Large Classes

The sheer number of students is the most prominent aspect of large classes, and it can also be a prominent advantage. However, size entails several other phenomena in large classes that instructors can exploit, including the diversity of the students in a large class, pooling of resources around the class, and the human reaction to membership in the class. Section 3 gives concrete examples and advice on how to exploit many of these advantages.

2.1 Diversity

Larger populations tend to present greater diversity across any scale. In large lecture classes, students have widely varied academic backgrounds, come from different cultures, and employ different problem solving styles. This diversity itself can be an important tool for an instructor.

Moreover, diversity engenders another valuable effect in large courses. For any given measure, a large class will tend to contain a larger number of students who substantially exceed the mean than a small class. Of course, these exceptional students would still be present were the large class divided into many smaller ones, but they would not participate together in a single community. The presence of this large "upper tail" in the classroom community is another advantageous aspect available to the instructor of a large lecture course.

2.2 Pooled Resources

Large classes usually have substantially greater instructional resources (including staff) available to them than small classes. While these resources may not be *proportionally* greater than those available to small classes, instructors can nonetheless gain advantages from these pooled resources.

A large staff can provide development capacity for the class that exceeds the proportional capacity of smaller classes. The staff can save time by sharing effort on code production, writeup, communication with the students, and grading standards that would otherwise be replicated across smaller classes.

Moreover, some of the advantages of diversity in large classes also accrue to large TA staffs. For example, despite small numbers of women in U. Washington CS's departmental TA pool, my students had three female role models among the twenty-six members of the TA and consulting staff.[2] These female staff members would still have been involved in separate, smaller classes, but at most three of those classes would have women on their staffs.

2.3 The Madness of the Crowds

> "Men, it has been well said, think in herds; it will be seen that they go mad in herds, while they only recover their senses slowly, and one by one." — Charles McKay, *Extraordinary Popular Delusions and the Madness of the Crowds* [6]

McKay describes how unscrupulous individuals and strange circumstances can push this herd mentality to astonishing results, but even a scrupulous instructor can turn this crowd atmosphere to pedagogical use. Roberts describes this force acting in the classroom, an infectious enthusiasm magnified by the scale of the large classroom that energizes students toward an activity [7]. This atmosphere can keep students engaged both with each other and with the material at hand.

3 Exploiting the Advantages

I tried to exploit the advantages described in the previous section in CSE142. The lessons I learned are listed below along with concrete examples from my class — and sometimes from the existing CS education literature — illustrating these lessons in practice. One overarching lesson was that *a positive outlook on large lecture courses helps keep the* instructor *positive about his course*.

3.1 Public Forums for Individual Effort

Capitalize on the crowd atmosphere and diversity of large classes by providing public opportunities for students to share unique or exceptional work.

Roberts has demonstrated that extra credit opportunities and contests have a positive impact in a large lecture course. He cites both the value to individuals who complete extra work as well as the whole class's avid interest in the results of these contests [7].

In my class, the last three programming assignments all used public forums to stimulate this interest and to encourage and showcase the class's diversity.

The students' third assignment (of five) was to write a program which would accept a series of words and, for each word, calculate its value in the base 36 number system. Then, the program would report to the user whether the result was prime or composite. We encouraged students to share examples of prime words which they discovered, and the students responded by posting examples to the class newsgroup.

[2] This count includes the staff of the parallel offering of the class. Outside of the weekly section, TA and consultant time was shared across the two offerings.

Initially, these were mostly amusing English words, but soon a Finnish student chimed in with a handful of Finnish (and French, Spanish, and German) primes. A Vietnamese student contributed a list of Vietnamese words which (when spelled without their accents) were prime in base 36.

In this assignment, the diversity of student background contributed to the excitement of the assignment. Indeed, while I had hoped that different students would find quite different sets of words, I never imagined that they would search in other languages!

The students' fourth assignment included a contest in which students selected the best of their peers' submissions (after the teaching staff had selected a small set of candidates). The assignment itself was to create a top-view, 2-D game in which the hero moves around and destroys enemies to advance to the next level. The top entries in the contest included multi-player versions of the game, games with a plethora of "power-ups," obstacles, and weapons, and one partially 3-D elevated-view game in which an animated elf slew hordes of animated skeletons (who, the game text explained, had stolen the elf's Mojo). The final round of the contest occurred in class, and the students voted by applause volume for their favorite game. This contest so captured students' interest that it was followed by a flurry of requests for the winners' code.

This assignment particularly exploited the crowd atmosphere of the class to generate excitement over programming artifacts.[3]

In the final assignment, pairs of students each wrote one room of a graphical MUD (Multi-User Dungeon). Each room had to include a puzzle and had to follow a simple API for multi-user play, but the assignment specification was otherwise intentionally vague. Code created by the TA staff stitched the individual rooms together and allowed students to log in to the MUD and control characters that could move through the rooms and interact with each other.

Many students commented in their feedback on the final assignment[4] that they enjoyed the opportunity to design their own programs in this assignment. Moreover, several students who viewed the MUD during my office hours expressed curiosity about how other students created their puzzles. Thus, the assignment used the diversity of a large class to create interest in design and alternate design styles.

On the other hand, about the same number of students commented in their feedback that the assignment was frustratingly open-ended. This dissatisfaction may have been caused by the tacit *requirement* for students to express their individuality in the assignment. Encouraging individual expression as extra credit rather than as a requirement may avert this dissatisfaction [7].

3.2 Setting the Tone

Use crowd mentality to encourage attitudes that improve the learning experience.

Daniel Klionsky suggests "setting the tone" of student participation by having all the students in class raise their hands on the first day [5]. He points out that even the shy students in the classroom are encouraged to join in because they raise their hands as part of a group, exploiting the crowd atmosphere. Ideally this same crowd atmosphere will encourage the continued practice of hand-raising.

I also used a first day exercise that leveraged crowd mentality. First, I told the students that this was a hard class but that *together* they were strong enough to overcome the difficulties. I told the students that I would demonstrate what strength lay in the hands of such a large community. Then, I screamed at the top of my lungs (a sound easily swallowed in a two hundred person lecture hall). Next, I asked the TAs to give their best yells (there were about six of them in the room). Finally, I asked the class to show us what they could do. The result dwarfed the staff's efforts (and was really, really loud). More importantly, it gave every student the opportunity to get used to speaking out in class and did it in a way that charged the atmosphere of the class.

3.3 Class Discussions

Capitalize on large "upper tails" and diversity to make class discussions effective.

I frequently used votes to keep students involved in the examples I presented. We would vote about whether loops would terminate, whether code would correctly implement an algorithm, or similar points. Although smaller classes can use this technique, large classes provide an exciting pedagogical advantage: students on both sides of an issue. In my experience of smaller classes, these votes often devolve into the entire class tentatively following one or two students who immediately support an answer — often the correct one. However, in my 200 student class, it was rare for a question to utterly lack eager supporters for any option. Thus, students were forced to think about the question during the 15 seconds or so that I gave them to commit. After the vote, I used the varied arguments students gave to justify their choices to illuminate the question from many different perspectives — dispelling preconceptions or building on different foundations of knowledge.

These votes exploit the diversity of a large class and the upper tails: a large class is more likely to supply the handful of assertive students with the varying viewpoints needed to start a voting block for each side of an issue. Once these stu-

[3] Moreover, both this assignment and the next relied on the staff's substantial development capacity.

[4] Every assignment included a writeup, and every writeup included a question asking what worked and what could be improved in the assignment.

dents raise their hands, others feel more comfortable joining them.

The students' second assignment also exploited the properties of a large class to generate discussion. The assignment was to simulate the "Let's Make a Deal" problem in which a contestant is offered a choice of three doors, one hiding a prize. After choosing, the host reveals one unchosen, losing door and asks if the contestant would like to switch to the remaining unchosen door or stay with her choice. The essential question of the assignment is: "how likely is the contestant to win if she switches?"

Although the problem is simple to state — and to solve, at least once you know the answer! — it has proven tricky to analyze even for people with training in probability. So, the students wrote a program to simulate the game and then automated the simulation and ran it many times to empirically determine the solution.

This assignment could have been given in a small class, but our large student body had the advantage of the substantial "upper tail" described above. In this case, that meant having enough students who were on the confident and assertive end of the spectrum to start a multi-sided argument on our class newsgroup over the probability of winning. The public argument ran over four days and involved 26 posts by six students expressing three distinct opinions: that the probability of winning if the contestant switches is 67% but 33% if she keeps her choice, that the probability is 50% in either case, and that it is 50% if she switches but 33% if she keeps her choice. Academic background played an important role in this argument; for example, one psychology major professed the last opinion based on his exposure to a psychological experiment with the "Let's Make a Deal" game that studied "belief perseverance." A CS major would be unlikely to construct the same argument (even if she reached the same result).

This argument added spice to the assignment. Better yet, it spawned another argument played out in staff office hours by many more students over how faithful the simulation of the game needed to be in order to prove the correct answer. This struck to the heart of the assignment's scientific subtext: when and to what extent we should trust computer simulations to decide theoretical questions. Once they were agreed on the correct answer, students argued hotly over whether simulating even "unnecessary" aspects of the game was important to obtain convincing results.

3.4 Human Algorithm Simulations

Exploit the large number of students to emphasize points in the analysis and execution of algorithms.

I often simulated algorithms using the students as data or processors. The large class size made demonstrations of the efficiency of algorithms compelling and provided enough raw material for simulation of randomized algorithms.

One exercise that taught students the basics of formulating and analyzing algorithms was to count the number of people in the classroom. We started by discarding the obvious algorithms of sounding off numbers or having me count them as too slow and then began brainstorming alternate algorithms (and critiquing their accuracy and speed). Students suggested a variety of algorithms including counting by rows, counting empty seats and subtracting from the room capacity, and checking the registrar's enrollment list.

Finally, we actually performed a randomized algorithm that I suggested in which every student was instructed to stand and think of a number between one and four and a color either red, green, or blue. All the students except the ones who chose two and red were instructed to sit back down. We then counted the number of standing students — which was now quite manageable — and multiplied the number by twelve to estimate the size of the class. This exercise leveraged the size of the class to make powerful points about algorithm analysis that would have been lost on a class of twenty.

(Interestingly, the exercise did suffer some problems in my class of 200: the final estimate of attendance was 48. After some discussion, the class decided that the low number resulted from people's preference for the number three over the other choices I had offered and that I should have used less familiar numbers.)

In another exercise, I used the size of the class to emphasize the value of short-circuit evaluation of logical "or." Again, every student was instructed to pick a value, this time either true or false. Then, I suggested that we calculate the "or" of all the students in class. I began traversing a row, student by student, collecting values to determine the final value. After a few students, I stopped and asked the rest how long it would take to finish all of them. We decided that it would take longer than we had left in the period. So, we discussed whether it was even necessary to continue and eventually settled on the correct answer: since someone had already chosen the value true, we could immediately conclude that the whole class's value was true.

Again, this exercise had a strong impact because of the class size.

3.5 Protecting Diversity

Where it does not contradict pedagogical goals, protect the diversity of your students and staff.

One unplanned benefit of large class size was the almost constant availability of help on the newsgroup. For example, in the argument over the "Let's Make a Deal" problem described above, three staff members also posted comments for a total of eight postings. One posted three comments between 8PM and 2AM, one posted four comments between 3AM and 8AM, and the last posted a single comment at about 2:30PM.

The diverse habits of the class's large instructional staff helped to ensure that urgent questions almost always received prompt answers. Although there is little I would have been willing to do to encourage this behavior, I did avoid discouraging it by making clear that TAs' time schedules were their own affair.

4 Conclusion and Challenge

I have described a novel point of view on the pedagogy of large lectures: making the best of their strengths as opposed to vitiating their weaknesses. Moreover, I have identified several potential strengths which instructors might exploit. As the examples from my CS1 class illustrate, it is possible to turn these strengths to practical, pedagogical advantage in the classroom.

Skeptical readers and instructors might be considering which of these techniques truly had positive impact on learning outcomes or which would transfer to their own classrooms. Unfortunately, I do not have direct quantitative answers to these questions.

Therefore, I challenge the CS education community to carry these initial anecdotes forward and formulate a more rigorous doctrine of teaching principles. Careful development and analysis of teaching techniques founded on the idea of magnifying the strengths of large courses will complement the existing educational literature and result in a more valuable experience for students. Moreover, this viewpoint will encourage instructors that are "stuck" with large lecture courses to strive for excellence rather than merely avoiding inferiority.

5 Acknowledgments

I thank people who provided help and discussion or who were involved in the teaching experiences described in this paper: Spring quarter 2001 CSE142 staff and students, Winter quarter 2000 CSE326 staff and students, and previous quarters' staffs; also, Richard Anderson, Martin Dickey, Michael Ernst, Andy Garland, David Kay, Rachel Pottinger, Jay and Shelley Wolfman, and the blind reviewers. This research was funded in part by an Intel Foundation Graduate Fellowship.

References

[1] Center for Excellence in Learning & Teaching, Penn State. "Forum on Large Classes." `http://www.psu.edu/celt/largeclass/forum.shtml`.

[2] Center for Research on Learning and Teaching, University of Michigan. "Teaching Strategies: Lectures and Large Classes." `http://www.crlt.umich.edu/tsllc.html`.

[3] Center for Teaching Excellence, University of Maryland. "Large Classes: A Teaching Guide." `http://www.inform.umd.edu/EdRes/FacRes/CTE/large/index.html`.

[4] Felder, R. M. Beating the Numbers Game: Effective Teaching in Large Classes. In *Proceedings of the 1997 American Society for Engineering Education Annual Conference* (Milwaukee, WI, June 1997).

[5] Klionsky, D. J. Tips for Using Questions in Large Classes. *The Teaching Professor*. `http://ase.tufts.edu/cte/occasional_papers/questions.htm`.

[6] McKay, C. *Extraordinary Popular Delusions And The Madness Of Crowds*. 1841. Public domain text: `http://www.litrix.com/madraven/madne001.htm`.

[7] Roberts, E. Strategies for Encouraging Individual Achievement in Introductory Computer Science Courses. In *Proceedings of the Thirty-first Annual SIGCSE Technical Symposium on Computer Science Education* (Austin, Texas, Mar. 2000), pp. 285–299.

[8] Washington Traditions. "The Wave." `http://www.fansonly.com/schools/wash/trads/020498aad.html`.