Teaching Challenge*

Sebastian Thrun
Stanford Artificial Intelligence Laboratory

This essay reflects my experience with the course "CS294: Projects in Artificial Intelligence," which I taught at Stanford in the Fall Quarter of 2004. The goal of the CS294 series is to deepen students' understanding of artificial intelligence through collaborative project work. What better idea than to focus the course on a true challenge: The *DARPA Grand Challenge*.

The DARPA Grand Challenge was a landmark event, launched by the U.S. Department of Defense. To win it, one had to build an autonomous robot capable of navigating 150 miles through punishing desert terrain in less than 10 hours, without human intervention. In March 2004, during the first Grand Challenge, 15 world-class teams had failed to achieve this goal, with the best-performing vehicle getting stuck only 5% into the course. So building such a robot was truly a challenge!

As was "teaching" this as a course.

CS294 was no common course. There was no textbook, no syllabus, no lecture. Instead, the course was about forming a team. It was about working together to build a robot in record time. Most students had never worked on robotics, and few had ever been part of a large team. So the course offered a totally new type of experience.

The course was open to all students on campus. From the nearly forty students who showed up on the first day of class, twenty chose to stay on. To manage such a large group of students, I divided the team into five groups, focusing on vehicle hardware, computing system, environment perception, motion planning, and low-level control. The first homework assignment was to have groups design their own work plan. Students had to come up with specifications of their contribution, a time line, and a sequence of milestones. The initial two weeks of the course were used for groups to develop this plan, and for negotiating interfaces with one another, so that all contributions would eventually work together. We jointly developed an interface document, along with a time line for the development process in form of a Gantt chart. The class as a whole had two joint milestones: an initial event in which

*This essay has been accepted for publication by the IEEE Robotics and Automation Magazine

Figure 1: CS294 in the Mojave desert, at starting chute of the 2004 DARPA Grand Challenge. After only two months of class work, the robot drove 8.5 miles along the course—further than any of the robots competing in 2004.

Stanley was to drive an autonomous mile on campus, and a final "autonomous desert mile" in the Mojave desert.

For many students, milestones and Gantt charts were uncharted territory. In fact, few had ever worked in a large team. After all, defining milestones and setting time lines is entirely different from solving homework assignments. But teams quickly converged to an initial Gantt chart full of subtasks and mini-milestones.

From the third week on, students focused on the technical work. In our class sessions, each group reported on progress, and we spent time resolving possible integration problems. In each session, I made a point of comparing each group's progress to our anticipated time line in the Gantt chart, so that we all could understand the rate of progress. In fact, the Gantt chart became my main method of spotting problems.

And problems were manifold. Over and over again, students proposed a technical idea, implemented it, and then observed it to fail in the field. Some students showed tremendous skills in identifying the underlying problems and solving them; others focused their energies on "explaining away" problems, or found reasons to conclude that the failure wasn't theirs. Conveying to students that as member of a team, it didn't matter who was at fault, required some efforts on my side. But in the end, most students got the message.

As the instructor, I worked hard to define and enforce deadlines, and to keep the team focused. I also tried to establish a systematic testing methodology. Students had to define performance measures and use them as debugging tools. Measuring

performance was entirely new to most students in class. And it was met with fierce resistance. It was as if I questioned the students' abilities to decide by themselves what works and what doesn't work. But as some of the better students noticed, this was the only way to spot bugs in the code, and to understand the side-effects of unsuspicious changes. I did not succeed in every instance to persuade the students to measure performance. But in those cases where I did, the result became noticeably stronger.

After many late evenings and weekends, the students managed to "deliver" the first integrated system just one week behind schedule. In mid-November, Stanley successfully drove its first mile, in many small loops around our garage. But many aspects of the system were still flawed. At times Stanley ignored obstacles, or simply drove off into random directions. So for the remainder of the course, students worked mostly with me and my research team on improving the basic system components. The course meetings became obsolete, since the students preferred to spend their time in the shop. Each group would merely send the least important member as a "delegate" to class. Instead, most interactions took place in our garage, where students now worked around the clock.

Finally, on December 1, 2004, we all traveled to Barstow, CA, where the 2004 Grand Challenge had taken place. Fig. 1 shows our group picture, taken at the starting line of the 2004 race. Stanley drove more like a drunken squirrel than a robot, but it was able to avoid obstacles and regulate its speed. 8.5 miles into the course, we finally were forced to intervene when the car failed to decelerate on a steep mountain descent. Still, the sense of achievement was tremendous. Stanley had gone further than any other robot had gone in the Grand Challenge (albeit at a lower speed). And we had built an end-to-end system in just eight weeks!

Grading this course turned out to be tricky. Most students were drawn into this project to the extent that they had dropped all other classes. Yet in grading, I decided to grade on contribution, not effort. Some students had shown great skill in directing their time towards the needs of the project, whereas others had payed little attention to all that. On a number of occasions I had to explain a non-perfect grade to a student who had poured all his time and enthusiasm into this project. Clearly, these are some of the least pleasurable moments of an instructor's life.

Months later, I heard back from several students that this course was among the most important experiences in their entire academic career. At Stanford, most courses take place in classroom settings with small, individual homework assignments. Students spend comparatively little time working in large teams focused on true challenge problems. I believe CS294 taught our students many valuable new lessons about all that. I am convinced those lessons will pay off, as our students leave academia and transition into a "real" world full of new grand challenges.