MATTHEW FISHER TEACHING STATEMENT

When designing a new course, I try to keep in mind that most students do not know exactly what field of study they want to pursue. They decide based on their experiences: if an introductory class is too hard or too boring, they are not likely to explore that field further. On the other hand, a well-designed class has the potential to energize students into pursuing a deeper interest in the subject. When developing my own courses, I am very conscious of the long-term effects a course can have and seek to make my classes approachable, challenging, and interesting — a goal which often requires achieving a very careful balance that can take many iterations to get right.

TEACHING EXPERIENCE

During my sophomore year at Caltech, I took a machine learning class taught by Yaser Abu-Mostafa, recipient of Caltech's Feynman Award for Excellence in Teaching. I was so impressed by his class and teaching style that I went to some length to become his teaching assistant for the same class my Junior year, even though undergraduates were generally not allowed to teach graduate classes. The following term, Yaser asked me to be the lead teaching assistant for the advanced version of the class. Both of these experiences helped me understand just how much effort goes into designing a course well. I still borrow many ideas from Yaser's teaching style, such as his beautifully organized lectures and his commitment to being available outside of class.

My core teaching experience is in computer graphics. Through a wide range of experiences including general discussions, teaching assistantships, and being the lead instructor, I have watched carefully as people from wildly different backgrounds approach graphics concepts. After each such interaction, I try to answer a series of questions:

- What topics did they find easy or hard?
- Do I need to present the topics in a different order so that things will make more sense?
- Are the assignments genuinely interesting, or are they doing them just because they have to?
- Do students understand how the concepts they are learning can be applied to real problems?

Based on these experiences, I work to continuously improve my teaching style to adapt to the changing dynamics of my students.

MENTORSHIP

Peer collaboration goes hand in hand with success in research: research cannot be done in a vacuum. Bringing more viewpoints to bear on a problem makes it more likely you will find an effective solution and helps you better explain your ideas so they can be well understood by the community at large. At Caltech and Stanford, I have sought out collaborations with computer scientists and media artists at a wide range of industrial and academic institutions. My collaborators help make sure that each paper presents a clear idea that solves a real problem.

Collaboration goes hand in hand with mentoring: introducing undergraduates to the concept of research, and bringing new graduate students up to speed with the research process, are both absolutely vital to the health of the community. Although all participants hold some responsibility for making sure the system works, ultimately those at the top are in the best position to understand and orchestrate the mentorship process. Within my own research groups, I am very conscious of this process and spend considerable time making sure that people are making the right connections. Without the right communication, students can get stuck working on frustrating or redundant problems that someone else already has worked through and understands well. With good communication, people feel less frustrated, research gets done more efficiently than students working alone, the less experienced students gain a better understanding of the research process, and the more experienced students gain important practice with how to explain their ideas concisely. I have greatly enjoyed being a part of this process at every level and look forward to developing such a community as a faculty member.

Collaboration is equally essential to the teaching process. Teachers must make sure that their teaching assistants understand the material and that they are able to pass that understanding on to students. Likewise, TAs need to be aware that through their more personal and direct interaction with students, they might learn aspects of how the course is going that the instructor does not. All the participants — the instructors, the TAs, and the students — need to work together to make sure that everyone is learning and having fun doing so. For all the classes I have been a part of, I consider it one of my first priorities to get the right kind of feedback and take the right actions to make this process go smoothly.

EXAMPLE COURSES

Undergraduate —

1. *Introductory Computer Graphics:* A practical course on the fundamental tools needed to understand and create graphics applications. The focus is on helping students interact with modern graphics pipelines and laying the foundations for more advanced graphics techniques. Topics include cameras, transforms, geometry, modeling, graphics hardware, shading languages, and illumination algorithms.

2. *Artificial Intelligence in Games:* Understanding and engineering intelligent agents for games and simulation. Emphasis is on gaining familiarity with the space of problems that might be encountered in real applications and how to adapt AI techniques to solve these tasks. Key algorithms include alpha-beta pruning, heuristic search, motion planning, and supervised learning.

3. *Practical Software Engineering:* Many students who start working on large projects with experienced developers are surprised at how much of the practice of software development they never learned. This course focuses on getting students experienced with the practical concepts that underlie software design, such as version control, debuggers, testing, bug-finding techniques, division of labor, and how to plan the design of large projects. Students work in large teams to gain practical experience with all aspects of the engineering process.

Graduate —

1. *Modern Graphics Techniques:* Current graphics applications often achieve exceptional quality rendering by using advanced algorithms that cannot be covered in an introductory course. This class focuses on bringing students up to speed with these techniques, including shadow mapping, voxel cone tracing, screen-space ambient occlusion, particle system simulation, and rigid body dynamics. Students work in teams on multi-week assignments to exercise course topics and gain experience on how to discover, implement, and adapt new techniques.

2. *Mathematical Methods for Graphics and Robotics:* Teaches the mathematics that underlie many of the most successful algorithms in graphics and related fields. Topics include numerical analysis, iterative and factorized solutions to linear systems, singular value decomposition, and forward and backward integrators. Emphasis is on demonstrating how the techniques can solve real and interesting problems.

3. *Research Topics in Computer Graphics:* This course covers a variety of topics at the forefront of computer graphics research. Topics include content generation, solid and fluid simulation, real-time global illumination, and motion synthesis. Students will gain familiarity with how to read, interpret, and implement recently published papers.