Alec Jacobson Teaching Statement

Computer science education is a unique blend of theoretical and applied work. This mixture demands thoughtful teaching supported by carefully planned curricula. My strategy is to design courses that intermingle traditional lectures and weekly homework with student-led discussions and project-driven evaluation.

My passion for teaching began early. Throughout high school and college, I tutored English to African immigrants. Later, while student teaching remedial and gifted-and-talented mathematics at a New York City public middle school, I realized the diversity of how students learn. Some students thrive off interactive lectures, while others do not fully digest a lecture until working out examples at home. Still others require one-on-one intervention.

During my graduate studies, I have enjoyed opportunities to continue teaching. As a teaching assistant at ETH Zurich, I designed homework and exams for the introductory computer graphics course and supplemented the course with a weekly recitation lecture. As a post-doc at ETH Zurich, I co-taught the Advanced Computer Graphics and Computer Vision Seminar.

Within the computer graphics community, I have taught several part-day courses. At the Symposium on Geometry Processing (SGP), I co-taught a course to budding students of the field on our award-winning C++ library for geometry processing, libigl. My ACM SIGGRAPH course on real-time shape deformation was invited for encores at ACM SIGGRAPH Asia and SGP.

Role-playing seminar

Seminars often suffer from poor attention retention. I experienced this while co-teaching a traditional paper-reading seminar. We attempted to salvage poor attention with “question quotas.” As the semester crept on, students had all but stopped reading the assigned material and presentations were disengaging.

When given the opportunity to teach a seminar on the same topic again, I was determined to improve student participation and satisfaction. With the help of Prof. Eitan Grinspun, I devised a weekly paper-reading seminar based on the concept of “role-playing.” All students would read the same paper each week, but each student would prepare different material for the class depending on their role that week. The roles would cycle throughout the semester, so each student played each role a few times while reading the dozen papers (see Appendix: Student Instructions).

Students latched on to the concept immediately and end-of-semester surveys were enthusiastic (see Appendix: Course Survey Quotes). Inspired on our success, colleagues at Carnegie Mellon University structured their seminar based on this curriculum.

Student Mentoring

Student mentoring has had a profound impact on my development as a computer science researcher—both as the student and as the mentor. I conducted undergraduate research at New York University under the mentorship of a then-PhD-student Yotam Gingold, now a professor at George Mason University and frequent collaborator. As a mentor myself, I have tried to mimic Yotam’s ability to put the student’s interests before his own. As a graduate student, I had the opportunity to mentor a talented undergraduate at ETH Zurich, Dingzeyu “Liding” Li. Liding enrolled as a PhD student at Columbia University the following year and has since published several papers in top-tier graphics journals. Leonardo Koller Sacht, whom I mentored during his student visit to ETH Zurich, is now faculty in Brazil. Our continued collaboration resulted in a top-tier graphic publication this past year.

Mentors learn alongside their students. Mentors must deconstruct their crystallized knowledge into its fundamentals before it is translatable to others. As faculty, I will not only commit to mentoring incoming students myself, but also commit to fostering an environment where our PhD students mentor younger students, undergraduates, and local high school students. I look forward to continuing my earlier outreach efforts to underrepresented demographics.
Proposed Courses

I am eager to teach the following courses:

**Geometry Processing (graduate)**
An advanced course studying the “life of a shape”: we cover a shape’s birth (surface reconstruction, geometric modeling), its numerical representation (meshes, splines, constructive solid geometry), its manipulation (smoothing, animation, editing) and its consumption (scientific visualization, 3D-printed fabrication).

**Computer Animation (undergraduate)**
An intermediate-level course, covering not only physically based simulation but also interactive techniques that broaden the impact of computer animation beyond cinema and video games to design, performance, and virtual reality. The course is driven by small projects culminating in a creative final project competition.

A project-oriented course, where students construct tools for state-of-the-art graphics and vision techniques by developing small apps throughout the course. Topics include convex optimization, high-performance data structures, parallel programming with the GPU, sparse linear algebra, and numerical integration.

**Introduction to Computer Graphics (undergraduate)**
A project-based course introducing students to the fundamentals of computer graphics built on the modern OpenGL graphics pipeline. Topics include rasterization, geometry representations, 2D image processing, global illumination and web-based graphics.

**Linear Algebra; Data Structures & Algorithms; Discrete Mathematics (undergraduate)**
Introductory courses on a variety of core computer science and applied mathematics topics. A course introducing students to: Linear Algebra (matrix operations, subspaces, singular value decomposition, Gram-Schmidt process), Data Structures & Algorithms (computational complexity, graph algorithms, dynamic programming, sorting), Discrete Mathematics (proofs, logic, combinatorics, set and graph theory).

**Computer Graphics Seminar (undergraduate or graduate)**
A paper-reading course thrusting students into the academic research literature via a weekly role-playing curriculum. Topics include computer animation, geometry processing, image processing, and computer vision, focusing on ground-breaking papers from SIGGRAPH in the last decades, but also touching on modern “hits.”

Appendix: Student Instructions

The following are the student instructions for the Role Playing Seminar discussed above.

This seminar is organized around the different “roles” students play each week: SIGGRAPH Reviewer, Archaeologist, PhD Student, Industry R&D Expert, Hacker, and Private Investigator.

**SIGGRAPH Reviewer:** Complete a full—critical but not necessarily negative—review of the paper. Answer all questions on the SIGGRAPH Review Form.

**Archaeologist:** Determine where this paper sits in the context of previous and subsequent work. Find and report on one older paper cited by the current paper and one newer paper citing this current paper.

**PhD Student:** Propose an imaginary follow-up project not just based on the current paper but only possible due to the existence and success of the current paper.

**Industry R&D Expert:** Convince us (your industry bosses) that it’s worth your time and out money to implement this paper into our company’s pipeline. Choose an appropriate company and product or application.

**Hacker:** Implement a small part of the paper or simplified version of the paper (e.g. 2D instead of 3D). Prepare a demo of your work for the class.

**Private Investigator:** Find out background information on one of the paper authors. Where have they worked? What did they study? What previous projects might have led to working on this one? We encourage you to contact the authors, but remember to be courteous, polite and on topic. Write that you’re in Prof. Grinspun and Dr. Jacobson’s seminar and include a link to this page.

**Everyone, every week:** Come with an alternative title and a missing result the paper could have included.
Appendix: Course Survey Quotes

The following are selected quotes from end-of-semester student evaluations of the Role Playing Seminar discussed above.

"The archaeologist role makes me think through what ideas this paper needed to succeed, and which other papers relied on this paper’s contributions to succeed."

"PhD student demands that you really appreciate the contribution and technical details of the paper. It makes you comb the current submission for assumptions, promises, or failure cases."

"This role [Industry R&D Expert] made me think of the return of investment of implementing the paper and of different ways in which the techniques in the paper could benefit the company as well as its products’ users"

"This role [Hacker] forced me to really think through the mathematics presented in the paper and to come up with clever alternatives when implementation details had been left out or when results were difficult to reproduce from certain sections."

"Being a PI is definitely fun because you get to follow one of the authors and look at the factors that influence the author’s interest and identity. I tend to view the paper as a by-product of the author’s identity and research interest."

"Reading only one main paper per class but with discussion through different lenses is definitely a really good approach for a seminar class like this because of two reasons: (1) you can cover a really good depth of a single paper instead of shuffling through 3–5 different papers with limited time, making the paper’s discussion more meaningful and interesting, and (2) everybody brings different things to the table that are coherent yet diverse in contexts."