Austin Henley — Teaching Statement

azhenley@memphis.edu

The opportunity to teach and mentor students is a principal factor in my decision to pursue being a professor. Great teachers have made an astounding impact on my career and have made a lasting impression on my life. It is because of these exceptional role models that I strive to go above and beyond in teaching, and to prepare students for whatever path they take next. My philosophy on teaching is that if you provide a student with the right scaffolding and learning environment, then they are bound to surpass all expectations.

My teaching experiences include being the Instructor of an undergraduate Operating Systems course of 37 students and being a TA of a Discrete Structures course. Not only did I enjoy teaching, but students were highly favorable of my energetic teaching style. I received a mean student evaluation score of 4.46/5 as Instructor (compared to the department's average of 4.26/5). To continuing growing as an educator, I aim to apply findings from **educational psychology** to my course design and to keep students engaged by relating concepts to **game development**.

1. Educational Psychology & Pedagogy

I have a substantial interest in educational psychology and the insights it can provide for the classroom. In fact, I graduated with a Cognitive Science graduate certificate with an emphasis on Educational Psychology. To achieve this, I took graduate-level psychology courses that covered topics such as cognitive load theory, active learning, the spacing effect, psycholinguistics, and reading comprehension. This has provided me with very applicable knowledge for the classroom. For example, I will present course materials to students three times spaced over a week and a half (e.g., initial lecture, small homework, and mini review lecture). This is based on evidence that suggests that repeated exposures that are spaced out over time will lead to better retention than if they had only seen the material in one long lecture with an immediate homework. Furthermore, I will carefully design study material and exam questions since eye-tracking studies have found that particular grammar structures (e.g., negation) have a dramatic effect on student performance.

I will also apply my knowledge of education research by leveraging recent pedagogical findings from the CS Education community, such as the Contributing Student Pedagogy and the flipped classroom technique. The Contributing Student Pedagogy encourages students to take part in each other's learning experience. This is a very collaborative approach, that incorporates peer review, open discussion, and group projects to enhance engagement and motivation. Furthermore, due to the difficult nature of debugging code, a flipped classroom where students read lectures as homework and do programming assignments in class has been found to be very beneficial. It allows students to get immediate help while working on coding problems rather getting stuck when they are outside of the classroom.

2. Relate Concepts to Game Development

A successful strategy I had while teaching Operating Systems was to relate abstract concepts to game development. For example, my students were struggling with page replacement algorithms so I framed the content in the context of managing entities in a game (e.g., 3d models, sound effects, etc.). After this lecture and homework assignment improved engagement tremendously, I began applying game development and other concrete examples to other lectures. Other examples include: teaching hardware interrupts by describing input polling and event messages in GUIs and teaching scheduling algorithms in the context of modern phone apps and game loops (i.e., draw, get input, synchronize network data, process, repeat).

The students were overwhelmingly favorable about these lectures. In fact, the students voted for me to re-teach garbage collection, which they had learned about in a previous course, but in the context of games. Moreover, several students contacted me outside of class to provide feedback on my lectures. A particularly motivating instance was when a female student spoke with me once after class to say that the concrete examples in my lectures are very helpful for her compared to the previous CS courses she had taken. Due to this great encouragement from my students, I plan to incorporate concrete and engaging examples in all of my lectures.

3. Proposed Courses

In the future, I look forward to teaching courses related to human-computer interaction and software engineering. Courses that I am excited to teach include user interface design & implementation, web development, game development, programming languages, introductory HCI, and introductory SE. Additionally, my experience has provided me the knowledge and proficiency to teach compilers, operating systems, research methods, and data structures. Moreover, I hope to create new graduate-level courses related to my research including human-centric development tools and empirical methods for lab studies. My aim for these research-oriented courses is to get students *doing* research that is relevant to their interests, not just reading about it.