3250: Instructional Strategies
ORID Reflection 2

Instruct

Active Learning in Action: Leveraging Neuroscientific Principles in Web Development Education

Author: Yuri Tricys
Date: May 05th, 2025

Objective

Chapter 3 of “Student Engagement Techniques”, Barkley & Major 2020, builds on several neuroscientific concepts, such as: schemata, transfer, association, memory, and retention to arrive at active learning and active learning techniques.

Active learning techniques are described in the following quote:

> *Active learning strategies or techniques, sometimes confusingly referred to as simply active learning, refers to several specific models of instruction, which range from more or less structured, including discussion, worksheets, cooperative and collaborative learning, discovery learning, experiential learning, problem-based learning, and inquiry-based learning.*
>
> *(Barkely & Major, 2020, page 41)*

This quote is interesting in the context of a web development curriculum because web development courses often involve live coding sessions, in which students participate with an instructor on their own computers.

Live programming sessions could engage students on multiple levels, including those used by Barkley & Major in their handbook for college faculty, to lead up to active learning techniques.

Reflective

I chose the notion of active learning techniques for this paper because in the past I’ve taken many courses, including multiple Front-End Masters courses taught by the likes of Kyle Simpson, Scott Moss, Brian Holt, and the ‘Primeagen’, that relied heavily on live coding (or recorded) coding sessions, which provided first hand experience in the ’active learning’ characteristics of live coding.

Live coding sessions feel like more dynamic and student-centered learning environments. They feel like active learning environments, that promote schemata development, and enhance the neuroscientific phenomina of transfer, association, memory, and retention.

A 2018 study by Raj et al. confirms that live coding enhances student engagement and learning:

> *We found that live-coding (1) makes the process of programming easy to understand for novice programmers, (2) helps students learn the process of debugging, and (3) exposes students to good programming practices. We also found that students prefer to code along with the instructor during a live-coding session rather than being mere observers.*
>
> *(Raj et al., 2018)*

Presumably, also, in person live coding sessions might help an instructor to identify areas where students are struggling and provide targeted support, helping to address knowledge gaps and reinforce understanding.

Interpretive

Live coding sessions encourage students to participate actively, think critically, and solve problems collaboratively. The discussion format and hands-on problem-solving experience could be considered ideal in terms of stimulating the development and refinement of schemata, which are mental frameworks that organize and structure knowledge.

An engaged and keen instructor could help students solidify their understanding of programming concepts by crafting feedback and guidance that matches his or her prior knowledge of student's learning patterns and experiences. Intelligent management of the discussion could facilitate rapid integration of programming knowledge into the student's existing schemata, through the associative process of linking new information to existing knowledge.

The concept of transfer, which refers to the ability to apply knowledge or skills learned in various contexts, can also be enhanced by the instructor's lesson planning and creative problem design. The instructor could, for example, present the problem in pseudo code, which is "extensively used in introductory programming courses to instruct computer science students in algorithm design" because it "lowers the [computer] language barrier students might face when learning to define algorithms" (Xu et al., 2024). Students could then be asked to solve the problem in an appropriate programming language, like TypeScript. Thus ensuring they make connections between theoretical concepts and practical applications across different computational contexts.

Moreover, when students actively participate in coding challenges, they're more likely to remember key concepts and retain information over time. The hands-on nature of live programming sessions helps encode information into long-term memory.

Decisional

Reflecting on live coding sessions in the context of neuroscience and active learning, though not new to me, is encouraging because it inspires me, as a future instructor, to move away from just 'traditional' lecturing methods toward well-designed live coding sessions.

In the past, I've worked as a curriculum writer for a company that delivered courses to industry across the US, in Canada, and in the United Kingdom. Amazon had awarded them the contract to deliver web development courses to Amazon non-technical staff, like warehouse employees and drivers. I designed the beginning to advanced JavaScript modules for that curriculum, which was more than 250 pages of course work. In the process I learned first hand how challenging and rewarding it can be to design problems for students. There is a lot of teaching that can be accomplished just in careful and clever problem design.

If an opportunity presents itself to apply similar curriculum design principles and procedures as those I developed first hand during that experience and those I've been exposed to during this course, I should find it both exciting and challenging to explore live coding problem design, especially if that design leads to the opportunity to deliver course material.

Conclusion

In conclusion, live programming sessions are an active learning technique that stimulate key neuroscientific processes, including schemata, transfer, association, memory, and retention. By providing students with hands-on experience in cleverly crafted coding challenges, live programming sessions can promote deeper understanding of concepts, but also improve retention, and enhance the ability to apply programming concepts to real-world problems, across platforms.

References

Barkley, E. F., & Major, C. H. (2020). Student engagement techniques: A handbook for college faculty (2nd ed.). Wiley Jossey-Bass.

Raj, A. G. S., Patel, J. M., Halverson, R., & Halverson, E. R. (2018, November 22). Role of live-coding in learning introductory programming. Department of Educational Leadership and Policy Analysis, University of Wisconsin-Madison. https://pages.cs.wisc.edu/~gerald/papers/LiveCoding.pdf

Xu, Z., Zhang, K., & Sheng, V. S. (2024). Logic error localization in student programming assignments using pseudocode and graph neural networks. *arXiv preprint arXiv:2410.21282*. https://doi.org/10.48550/arXiv.2410.21282

AI Models Used in Report

The ideas, structure, writing, and editing in this paper were performed by the author. Various AI models were used in collecting data, verifying data, and formatting various arguments.

AI models used:

- LLama 4 Maverick
- LLaMA 2 LLM Chatbot
- Claude 3.7 Sonnet
- Qwen 14B
- Llama 3.1 8B
- Mixtral 8x7B