Visualizing Classic Synchronization Problems

Elizabeth Koning, Christiaan Hazlett, Dr. Joel Adams. Calvin College, Grand Rapids, Michigan

Problem and Motivation

Classic synchronization problems are commonly used to introduce CS students to concurrency and synchronization.

Operating Systems textbooks use figures and diagrams, but these are limited the static nature of the paper on which they are printed.

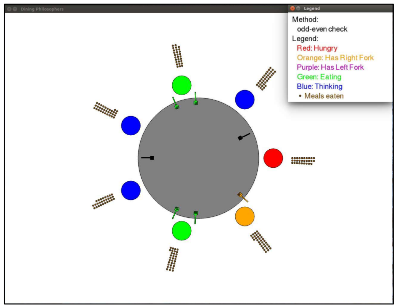
By contrast, software is a more flexible medium that can be used to create dynamic, interactive visualizations of these problems.

Uniqueness

Our visualizations differ from previous, related work (e.g., [2]) through its:

- Focus on **parallel algorithms**, as opposed to sequential ones (e.g., [3]).
- **Real-time execution** & visualization instead of "post-mortem" playback.
- Visualization of concurrent entities instead of synchronization primitives.
- Usage of the **Thread Safe Graphics Library** [1] as opposed to GTK, which is not thread-safe.

Approach and Visualizations

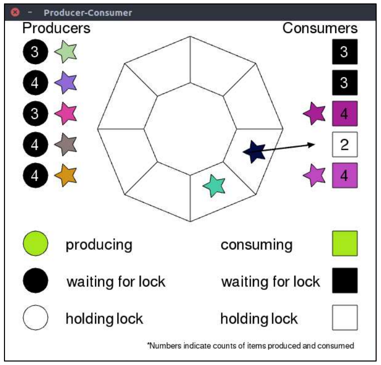


Method:
odd-even check
Legend:
Red: Hungry
Orange: Has Right Fork
Purple: Has Left Fork
Green: Eating
Blue: Thinking
• Meals eaten

Figure 1: *Dining Philosophers*

The Dining Philosophers, Producers-Consumers, and Readers-Writers Problems are three "classic" problems that are most commonly-used by textbook authors. Our project focused on visualizing these three problems and their solutions.

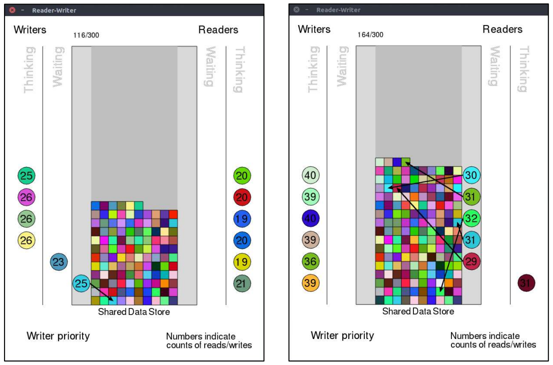
For each of the problems, our visualizations allow customization with number of threads and choice of algorithm.



producing
waiting for lock
holding lock
consuming
waiting for lock
holding lock

*Numbers indicate counts of items produced and consumed

Figure 2: *Producers-Consumers*



Writer priority

Numbers indicate counts of reads/writes

Writer priority

Numbers indicate counts of reads/writes

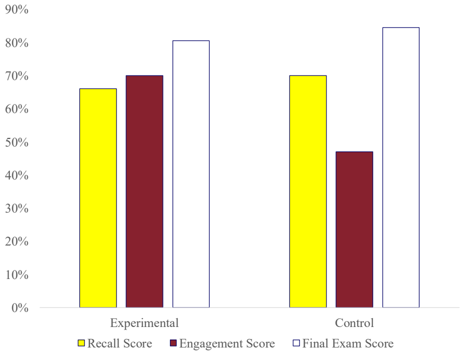
Figure 3: *Readers-Writers*

Experiment and Results

To assess our visualizations, we identified two research questions:

- **RQ1**: Does interacting with our visualizations improve long-term recall of these problems, compared to reading a textbook?
- **RQ2**: Do students find our visualizations to be a more engaging way to explore these problems, compared to reading a textbook?

To answer these questions, we ran an experiment comparing a control group reading textbooks to a treatment group interacting with our visualizations (N=22 students). Two weeks later, both groups took an online quiz about the problems and rated their engagement.



	Treatment	Control	P-value
Recall Score	9.9	10.5	0.3387
Engagement Score	4.2	2.8	0.0090
Final Exam Score	80.5	84.5	

Conclusions

The results of our experiment indicate:

- No significant difference in learning occurred between students using our visualizations and those reading textbook materials.
- Students interacting with our visualizations were significantly more engaged than those reading textbooks.
- Our randomized control and treatment groups were academically imbalanced.

Our visualizations, videos, and quizzes are available upon request.

Full source code available at:

github.com/Calvin-CS/TSGL

References

[1] J. Adams, P. Crain, C. Dilley, S. Nelesen, J. Unger, M. Vander Stel, Seeing Is Believing: Helping Students Visualize Multithreaded Behavior, *Proc. of 47th ACM Technical Symposium on Computer Science Education* (SIGCSE'16), March 2016, 473-478. DOI= 10.1145/2839509.2844557.

[2] S. Carr, J. Mayo, and C.K. Shene, ThreadMentor: a Pedagogical Tool for Multithreaded Programming, Journal on Educational Resources in Computing (JERIC), 3(1), March 2003, Article 1. DOI= 10.1145/958795.958796.

[3] E. Fouh, M. Akbar, C. Shaffer. The Role of Visualization in Computer Science Education. *Computers in the Schools: Interdisciplinary Journal of Practice, Theory, and Applied Research*, Vol. 29, 2012, 95-117. DOI=10.1080/07380569.2012.651422.

Acknowledgements

We would like to thank the Calvin College Science Division and the Calvin Research Fellows program, whose support made this work possible.