Games, computer science, and education

Peter Drake

Published online: 20 Apr 2013.

Submit your article to this journal ↗

View related articles ↗

EDITORIAL

Games, computer science, and education

Games and computer science are kindred spirits. Both start with a few simple components and rules, from which are built towering structures of majestic complexity. The programmer and the player of Chess or Go savor very similar challenges; they experience the same feeling of triumph upon discovering a clever solution. Many – perhaps most – computer scientists grew up sharpening their brains on games. Computer science has returned the favor, providing the foundations for an extremely robust electronic game industry.

Games and education also go together. Suits defined playing a game as "the voluntary attempt to overcome unnecessary obstacles" (1978). Is this not the ethos of the purest academic inquiry, which tackles questions precisely because the answers are unknown? More practically, we can use games to teach (Drake & Sung, 2011), teach about creating games, and even "gamify" the entire educational experience (Sheldon, 2012).

For the relationship between computer science and education, the reader is referred to all previous issues of this journal.

This special issue presents five papers exploring the many areas where games, computer science, and education overlap.

Schäfer et al. present a new game for teaching mathematical logic. Their game covers the resolution algorithm, which might appear in a course in mathematical logic or artificial intelligence. Educational games, a subset of serious games, have come a long way since the tedious Trivial Pursuit variants of our childhoods. This one features several modes, allowing one or more players to harness the intrinsic motivating power of games.

The next two papers shift from games built for students to games built by students. There are many courses where students build games on top of premade game engines. Such courses allow students to experience the thrill of creating a good-looking game without writing every aspect of the program (graphics, physics, etc.). Claypool turns this approach on its head by having the students write the game engine, providing his students with a deep understanding of the structure of such a program.

Kurkovsky describes curricular modules for game design assignments focusing on mobile devices. He also discusses the motivation for and evaluation of these modules.

McGill, Settle, and Decker present a study of undergraduates at four institutions offering degrees in game design. They compare the results of

their survey with previous work on computer science students and on workers in the game industry.

Capt. Adrian A. de Freitas and Michelle M. de Freitas conclude the issue with Classroom Live, a new tool for managing a gamified classroom. The challenges encountered and lessons learned while developing and deploying this tool will be helpful to other educators exploring this innovative approach.

The editor wishes to thank the reviewers for their hard work and prompt, thorough reviews. The reviewers included are Judy Cushing, Michele Dijkstra, Daniel Ford, Mark Goadrich, Scott Wallace, Karen Ward, and Richard Weiss. The editor also thanks the authors for their contributions to this explosively growing area of research.

Peter Drake

Department of Mathematical Sciences, Lewis & Clark College
Portland, Oregon, USA
drake@lclark.edu

References

Drake, P., & Sung, K. (2011). Teaching introductory programming with popular board games. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education (SIGCSE 2011)*, Dallas, TX.

Sheldon, L. (1012). *The multiplayer classroom: Designing coursework as a game*. Boston, MA: Cengage.

Suits, B. (1978). *The grasshopper.* Cited in McGonigal, J. (2011). *Reality is broken: Why games make us better and how they can change the world*. New York, NY: Penguin.