Scalable Game Design for Middle School (Case Study 6)

An Engaging Way to Introduce Computing

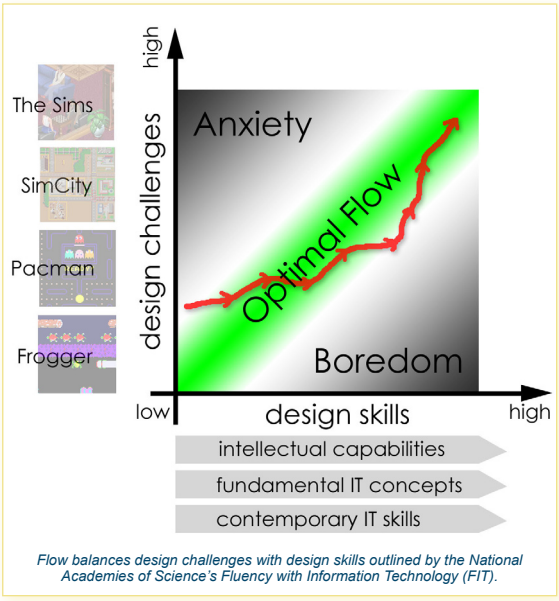
K-12 Education

It's not so easy to build and design a working video game, but a well-crafted learning environment makes it possible and interesting for many students. The middle school computing curriculum in Colorado's Boulder Valley School District (BVSD) uses Scalable Game Design to introduce computer programming in engaging ways and helps students develop IT skills aligned with ISTE'S National Educational Technology Standard of Creativity and Innovation.

In the very first lesson, students make their own Frogger-like game to publish on the web. Over the course of a one- to two-month module, students learn more sophisticated topics in order to create increasingly complex games and computational science applications. According to Len Scrogan, Director of Instructional Technology for BVSD, the results of the BVSD implementation include motivated students, engaged teachers, and excited parents.

THE THEORY BEHIND THE CURRICULUM

Scalable Game Design uses the premise that learning is most successful when students engage in tasks that are difficult enough to be interesting but not so difficult that they become frustrating. The psychological notion of "flow" can help manage this tension. This notion suggests that students learn best when in "optimal flow", where design challenges match design skills and anxiety is relatively low. In this stage, students are highly receptive to guided learning even if the topic appears too difficult. Scaffolding lessons this way helps students progress from simple arcade games to games that require sophisticated artificial intelligence.



Flow balances design challenges with design skills outlined by the National Academies of Science's Fluency with Information Technology (FIT).

EVALUATION: BOTH GIRLS AND BOYS MOTIVATED TO PROGRAM

AgentSheets, a scalable game design product, has been evaluated in two small studies for its effectiveness in motivating middle-school students to learn programming. In a summer elective course, 36 middle school boys and girls used AgentSheets to experiment with programming concepts and create games or animations. Interestingly, while most students expressed a desire to continue with AgentSheets, students with low-technology experience expressed a slightly stronger desire than those with high-technology experience. By the end of the course, girls and boys also expressed similar levels of desire to continue using AgentSheets. Another study using AgentCubes, a 3D simulation and programming tool developed by the creators of AgentSheets, found that all students were able to create a working 3D game in less than five hours. This study was conducted in an afterschool program that included girls, inner-city low-income students, and students in a U.S. technology hub. All students performed well in developing and troubleshooting their creations.

CHARACTERISTICS OF A SUCCESSFUL EDUCATIONAL PROGRAMMING ENVIRONMENT

- Accessible to students without prior programming experience
- Simple enough to make a working game in three hours or less
- Powerful enough to allow implementation of sophisticated artificial intelligence algorithms
- Works for game and computational science applications
- Transitions to traditional programming such as Java

So far, one product on the market combines these ingredients, AgentSheets. Originally developed at the University of Colorado, AgentSheets is available as a ten day free trial at www.agentsheets.com.

RESOURCES

For more information on AgentSheets and related resources, see www.agentsheets.com. AgentSheets is funded by NSF.

Ioannidou, A., Repenning, A. and Webb, D. (2008). *Using Scalable Game Design to Promote 3D Fluency: Assessing the AgentCubes Incremental 3D End-User Development Framework*. Paper resented at the 2008 IEEE Symposium on Visual Languages and Human-Centric Computing, Herrsching am Ammersee, Germany.

Repenning, A, & Ioannidou, A. (2008). Broadening participation through scalable game design. *SIGCSE Bulletin (40)*1, 305-309.

Walter, S.E., Forssell, K, Barron, B, & Martin, C. (2007). Continuing motivation for game design. *CHI '07 Extended Abstracts on Human Factors in Computing Systems*, 2735-2740.

Case Study Contributors: Alexander Repenning, Alexander.Repenning@colorado.edu, and Len Scrogan, len.scrogan@bvsd.org.

NCWIT offers practices for increasing and benefiting from gender diversity in IT at the K-12, undergraduate, graduate, and career levels.

This case study describes a research-inspired practice that may need further evaluation. Try it, and let us know your results.

ncwit.org **Authors | Catherine Ashcraft and Stephanie Hamilton**

NCWIT Investment Partners: National Science Foundation, Avaya, Microsoft, Pfizer, and Bank of America

How Do You Introduce Computing in an Engaging Way?

with Case Study 6

K-12 Education

Experience with computers between boys and girls has equalized, but boys continue to have greater knowledge of computing and programming *concepts* than do girls. Not so in biology, chemistry, or mathematics, where both boys and girls are encouraged to provide evidence of proficiency when they apply to college. High school study of these subjects familiarizes students with the content and concepts, and gives them confidence. The result is that women's undergraduate completion rates have neared parity in these disciplines.

Because IT study is elective in almost all K-12 schools, developing relevant and interesting assignments that appeal to a broader audience is recommended for:

- fostering a climate where the non-predisposed can belong both academically and socially
- recruiting students who are not predisposed to pursuing computing
- exposing fundamental computing concepts to inexperienced learners

Is prior programming experience required for students to be successful in an IT program? Most undergraduate departments would say no. That is, experience with programming is not the same as expertise in problem-solving, algorithmic thinking, or computing theory. Yet research shows that introductory courses and their embedded assignments work better for students who have *some* experience with programming.

Research also shows that students with programming experience are more confident and more successful in introductory courses than are their inexperienced peers. Students with lower grades or less confidence are less likely to persist in an IT major. What is more, when introductory courses have limited opportunities for talking to other students (e.g., collaborative learning), inexperienced students have little information on which to judge whether they belong academically in the major. Hence more women than men switch out of IT majors (most often to other sciences or mathematics).

MAKING IT MEANINGFUL

Educational researchers emphasize the importance of linking educational materials and curricular programs to students' existing knowledge and experiences. When class syllabi list topics and assignments that focus on unfamiliar concepts with limited, if any, relationship to a student's life experience or interests, she or he is unlikely to take that class. High school curricula contribute to low enrollments in college computing because, under the existing educational policy of election, computing is rarely required in secondary schools. This means that students are likely to have a narrow and inaccurate view of what IT study involves, what careers are possible, or what kind of people "do" IT. Given the very small proportion of females who study computing in high school, females are less likely to choose IT in college.

The challenge to educators at all levels is to develop engaging assignments and curriculum that can appeal to a variety of students with different learning styles, interests, socio-cultural backgrounds, and abilities, while maintaining the rigor of the discipline. Putting the concepts of computing in appealing contexts and building on existing competence can both reduce entry barriers and level the playing field for those with limited experience.

Creative assignments that teach algorithmic thinking while also calling on students' existing knowledge or interests, may serve to both recruit and retain students. When experienced and inexperienced students use non-computer-based assignments to learn computing concepts, they quickly realize that their peers with programming experience are not necessarily better at algorithmic thinking, just more experienced with programming. Building confidence through relevant and interesting assignments is a promising practice for motivating student enrollment and retention.

RESOURCES

Lecia Barker and William Aspray, "The State of Research on Pre-College Experiences of Girls with Information Technology." In McGrath Cohoon, J. and W. Aspray (Eds.) *Women and Information Technology: Research on the Reasons for Under-Representation*. Cambridge, MA: MIT Press, 2006.

Joanne McGrath Cohoon and William Aspray, "A Critical Review of the Research on Women's Participation in Postsecondary Computing Education." In McGrath Cohoon, J. and W. Aspray (Eds.) *Women and Information Technology: Research on the Reasons for Under-Representation*. Cambridge, MA: MIT Press, 2006.

Authors | Lecia Barker and J. McGrath Cohoon